Benching the New Solaris VM

REVIEWS
- Visual J++ 6.0
- SilverStream 2.0
- JaVISION 1.0

JAVA Pro™

www.java-pro.com
A Fawcette Technical Publication
March 1999, Vol. 3, No. 3

SECURITY for SERVLETS

Java-Powered Databases

Give Your Apps Some Muscle

Simplify DB Apps with **Objects**

Build Your Own **Portal Site**

Getting Started with **XML**

Memory-Mapped Files for Java

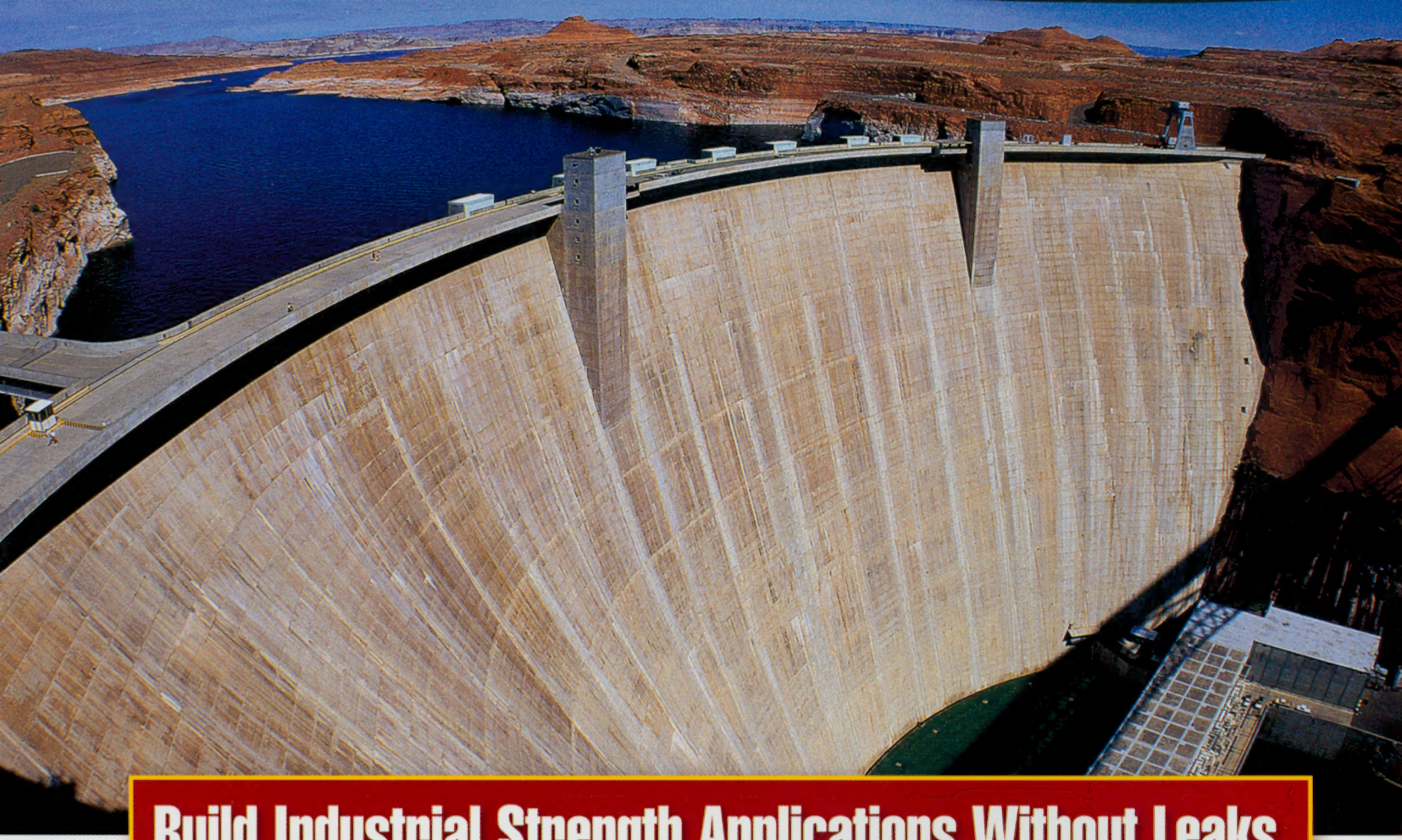
Tips on Semaphores

Code Online
Free Source Code
www.java-pro.com

U.S. $5.95 (Canada $7.95)

03>

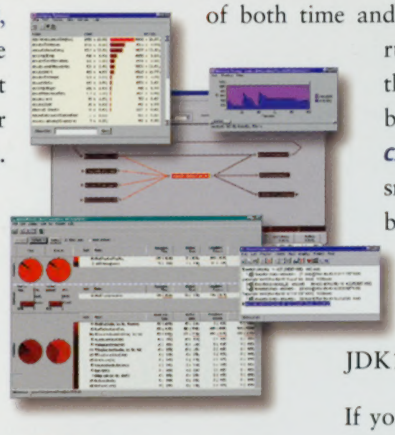
7 25274 91607 8



Build Industrial Strength Applications Without Leaks

Whether you're building a high performance hydro-electric dam or a ***MISSION-CRITICAL JAVA APPLICATION***, you know that those pesky leaks can sabotage everything! You need a professional tool that is accurate, easy to use and helps you deliver ***SCALABLE, ROBUST PERFORMANCE*** from start to finish.

JPROBE PROFILER is used by Java developers around the world to improve the performance of their Java applications. And now, the most ***POWERFUL TOOL*** for finding and eliminating Java performance bottlenecks makes it easy for you to find and plug memory leaks in your Java code. Use the new ***HEAP REFERENCE GRAPH*** to instantly see a heap snapshot analysis and untangle even the most complex object references that prevent garbage collection.



JPROBE Profiler

JProbe Profiler gives you the most ***ACCURATE MEASUREMENT*** of both time and memory consumption while your program runs. It tracks everything for you, thread by thread, with method details shown clearly right back to each line of source code*. The advanced ***CALL GRAPH*** helps you analyze your profiling snapshot and quickly drill-down to performance bottlenecks. And you'll get superior performance where you want it because JProbe Profiler provides ***MULTI-PLATFORM*** support for Windows NT and Solaris in JDK1.1, JDK1.2, and JIT environments.

If you develop Java code for a living, you need a ***PROFESSIONAL JAVA PROFILER.*** Get a free evaluation copy of ***JPROBE PROFILER*** right now and start making your Java code industrial strength, without leaks.

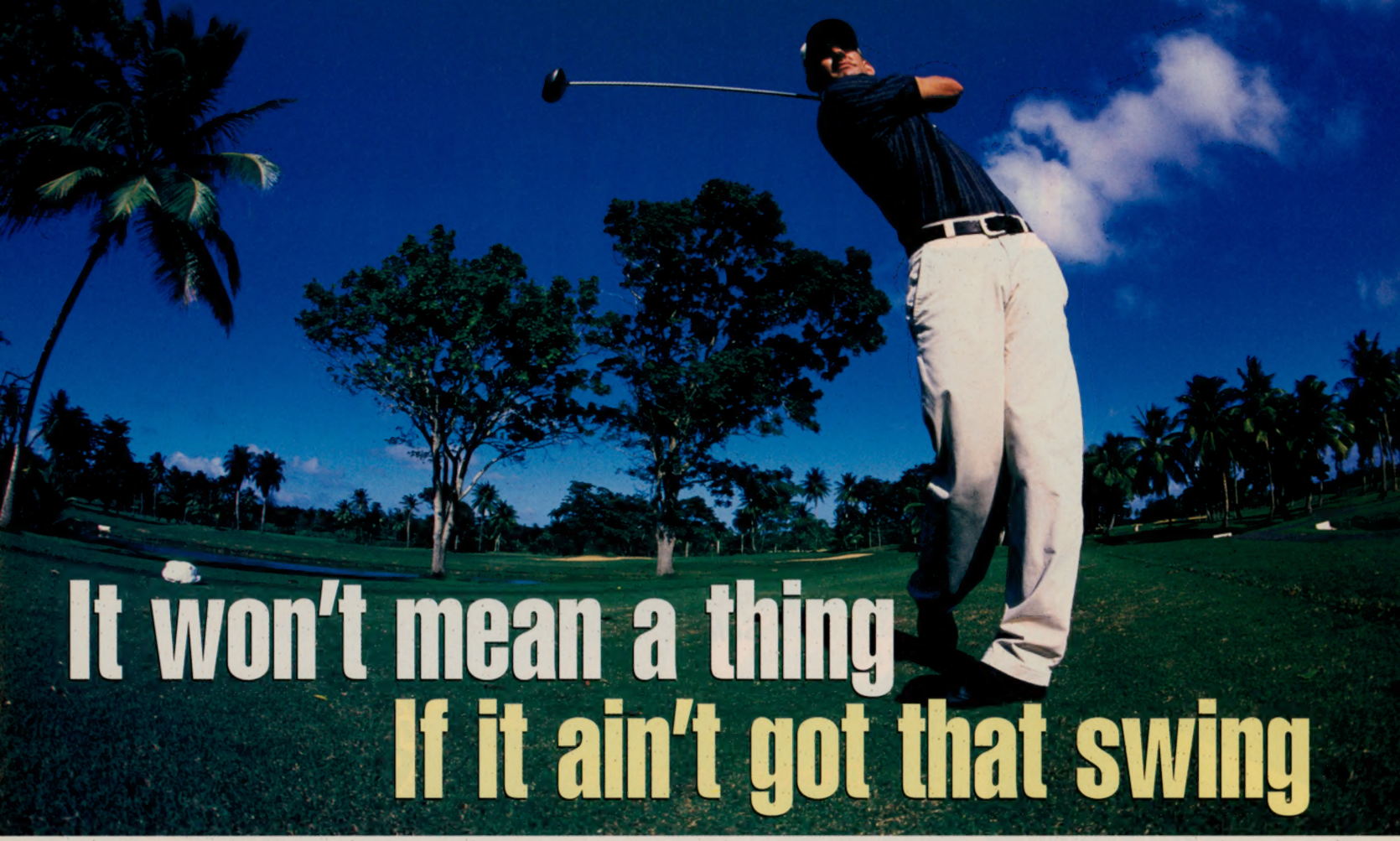
Try JProbe Now! www.klg.com/jprobe | ***The Easiest Way to Search and Destroy Java Performance Bottlenecks***

KL GROUP
Software Development Productivity

KL Group Inc. Toll-Free: 1-888-328-9599 Tel: 416-594-1026 Fax: 416-594-1919
KL Group Europe B.V. Tel: +31 (0)20 679 95 03 Fax: +31 (0)20 470 03 26
www.klg.com

JAVA COMPATIBLE

* Source code not required to profile with JProbe. JProbe is a trademark of KL Group Inc. All other products are trademarks or registered trademarks of their respective companies. JP003



It won't mean a thing
If it ain't got that swing

JCLASS JavaBeans

You need ***powerful components*** for professional Java development. But if they don't work with Swing and the latest JDK 1.2 release... they won't mean a thing.

JClass, the world's most popular suite of 100% Pure JavaBeans, now includes ***JClass SwingSuite***, the essential set of extensions and enhancements for Swing in JDK 1.2. And with the power of JClass Suites, you'll be building for the future ***today.***

JCLASS WORKS WITH:

✓ JDK 1.0.2 (AWT)	✓ IBM VisualAge for Java	✓ Progress Apptivity
✓ JDK 1.1 (AWT or Swing)	✓ IST Visaj	✓ SilverStream
✓ JDK 1.2 (Swing)	✓ Microsoft Visual J++	✓ Sun Java WorkShop
✓ Bluestone Sapphire/Web	✓ NetBeans	✓ SuperCede
✓ Borland JBuilder	✓ Powersoft PowerJ	✓ Symantec Visual Café

New **JClass SwingSuite** - easily add the power of MDI, wizards and a robust collection of thread-friendly, lightweight components including JCTreeExplorer, JCTreeTable, JCSpinBoxes, a configurable DateChooser control and more!

- **JClass LiveTable** - the high performance Java grid/table for static, dynamic or db-bound data
- **JClass Chart** - the ultimate in Java charting and graphing, designed for superior performance
- **JClass HiGrid** - the innovative RAD outline-grid for hierarchical dynamic data
- **JClass DataSource** - a robust, hierarchical, multiple-platform data source
- **JClass Field** - complete input and validation for popular data types including calendar

Need rock-solid, robust Java GUIs? Professional Java developers around the world have made the award-winning JClass family of components their choice. Where else will you find seamless IDE integration and support for all major JDK versions? ***Try JClass today for great performance with an improved swing!***

EVALUATE/ORDER ONLINE www.klgroup.com

KL GROUP
Software Development Productivity

KL Group Inc. Toll-Free: 1-888-328-9598 Tel: 416-594-1026 Fax: 416-594-1919

KL Group Europe B.V. Tel: +31 (0)20 679 95 03 Fax: +31 (0)20 470 03 26

www.klgroup.com

* Databinding feature requires purchase of JClass DataSource, or use of Borland JBuilder Professional or C/S edition, or Symantec Café dbDE edition. JClass, JClass Chart, JClass DataSource, JClass Field, JClass HiGrid, JClass LiveTable and JClass SwingSuite are trademarks of KL Group Inc. All other product names are trademarks or registered trademarks of their respective owners. JP004

Zero G Software's

InstallAnywhere™ 2.5

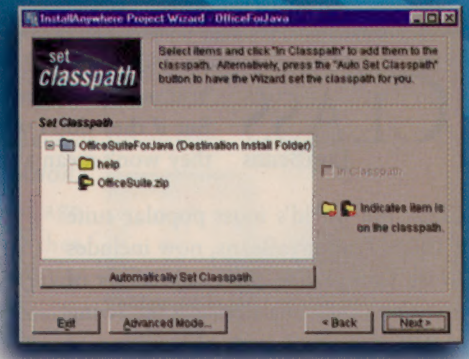
New Features Include:

- SpeedFolders™ - faster-than-native installations
- Java 2 support
- Optimized for large installations
- Install with or without a Java™ VM
- LaunchAnywhere™ applications launchers for:
  Windows 95, 98, NT, Mac OS, OS/2,
  Solaris, Linux, HP-UX, IRIX and AIX

Professional Look and Feel

Installers have a professional look and feel that's familiar to all users on all platforms. You can run installers from CD-ROM or directly from the Web.

Create Installers Quickly

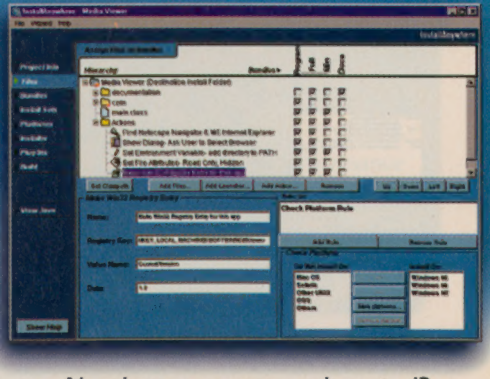

InstallAnywhere Project Wizard - OfficeForJava
set classpath
Select items and click "In Classpath" to add them to the classpath. Alternatively, press the "Auto Set Classpath" button to have the Wizard set the classpath for you.
Set Classpath
OfficeSuiteForJava (Destination Install Folder)
help
OfficeSuite.zip
In Classpath
Indicates item is on the classpath.
Automatically Set Classpath
Exit
Advanced Mode...
< Back
Next >

In six steps you can create an installer with the Project Wizard. It sets the classpath and finds "main" classes automatically.

JAVA DEVELOPER'S JOURNAL JDJ EDITOR'S CHOICE AWARD

Editors' Choice JavaWorld 1998 WINNER

Power and Control



Need more power and control? The Advanced Designer lets you control everything! Create custom install sets, icons, installation rules, multiple destinations and more...

Standard Edition $995

SIX FREE LANGUAGES! Build localized installers in English, French, German, Italian, Japanese, and Spanish

Express Edition $495

New Optional Add-ons:
- Western Language Pack
- Eastern Language Pack
- Advanced Actions Pack

zero SOFTWARE

FREE TRIAL VERSION available at www.ZeroG.com

Zero G Software 118 King Street, Suite 415 San Francisco, CA 94107-1916 Phone: (415) 512-7771 Fax: (415) 512-7775

JAVA Pro™

March 1999,
Vol. 3, No. 3

Contents

Features

20 Java in the Database
by Ken North
As Java gains momentum in the enterprise, Java developers are becoming more sophisticated in their use of the corporate database. Here's a rundown on the state of the art, from JDBC and simple SQL queries to more advanced online-processing applications, including a look at Java as a stored procedure language.

28 Using Memory-Mapped Files in Java
by Tom Guinther
Tom continues his work on bringing Win32 functionality to relatively pure Java code. This month, he demonstrates how to share data between apps by the judicious mix of a base class, JNI code, and C++.

34 Security for Servlets
by Jason Hunter with William Crawford
Web Security is just as important for servlets as for any other server-side code. Explore your options from simple HTTP authentication to digital certificates and the Secure Sockets Layer.

48 An Introduction to XML for Java Programmers
by Piroz Mohseni
Java101 Like Java, XML has emerged from an early hype phase to become a valuable tool in handling data. Learn how Java can make an XML document's data and metadata come to life in your apps.

Reviews

10 Visual J++ 6.0
12 SilverStream 2.0
14 JaVISION 1.0

Columns

54 Ask the Java Pro
by Daniel Savarese
Daniel addresses some of the real and perceived inefficiencies of Java and highlights some of the changes of the new Java 2 release.

62 Javatecture
by James Cooper
Good designs require more than just using objects, they require thinking in objects. James walks through a database application and shows how objects can simplify an application.

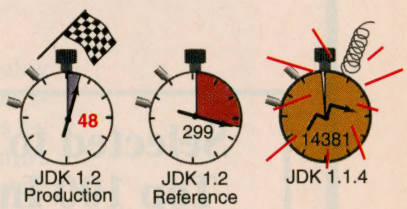
66 Impure Thoughts
by Rick Grehan
Java's mixed blessings strike again. It's easy to create a semaphore, but will it behave the way you want it to?

72 Servlet Solutions
by Larry O'Brien
Web Take on the dark sorcerers of Yahoo! with a servlet-driven portal site and Java Server Pages.

78 VM Roadtest
by Paul Tyma
Sun's new Solaris VM sports numerous improvements—the result is top-tier Java performance.

Departments

5 Editor's Note
by Tyler Sperry
7 Letters
16 New Products
38 Readers Survey
61 Ad Index



Find out if Sun's new VM takes the pole position; see page **78**.

Java Pro is published monthly for $34.97 per year, by Fawcette Technical Publications, 209 Hamilton Avenue, Palo Alto, CA, USA 94301-2500. Tel. 650-833-7100; Fax. 650-853-0230. **Postmaster:** Send address changes to Java Pro, P.O. Box 54584, Boulder, CO 80322-4584. Application to mail at Periodicals Postage Rates is pending at Palo Alto, CA and additional mailing offices. © **1999 Fawcette Technical Publications**, all rights reserved. *Java Pro* is an independent publication not affiliated with Sun Microsystems. Sun Microsystems is not responsible in any way for the editorial policy or other contents of the publication. All contents of *Java Pro* are copyright © 1999 by Fawcette Technical Publications, unless otherwise noted. "VBITS" and "Interactive Developer" are trademarks of Fawcette Technical Publications, a California Corporation, James E. Fawcette, President. Java is a trademark of Sun Microsystems. Rather than put a trademark symbol in every occurrence of other trademarked names, we state that we are using the names only in an editorial fashion with no intention of infringement of the trademark. Although all reasonable attempts are made to ensure accuracy, the publisher does not assume any liability for errors or omissions anywhere in the publication. **Canadian GST#: 86831 0764 ISSN: 1096-4495.**

Web These features focus on Java on the Internet.
Java101 New to Java? These features help get you going.

Serious "Java to SQL" Connectivity

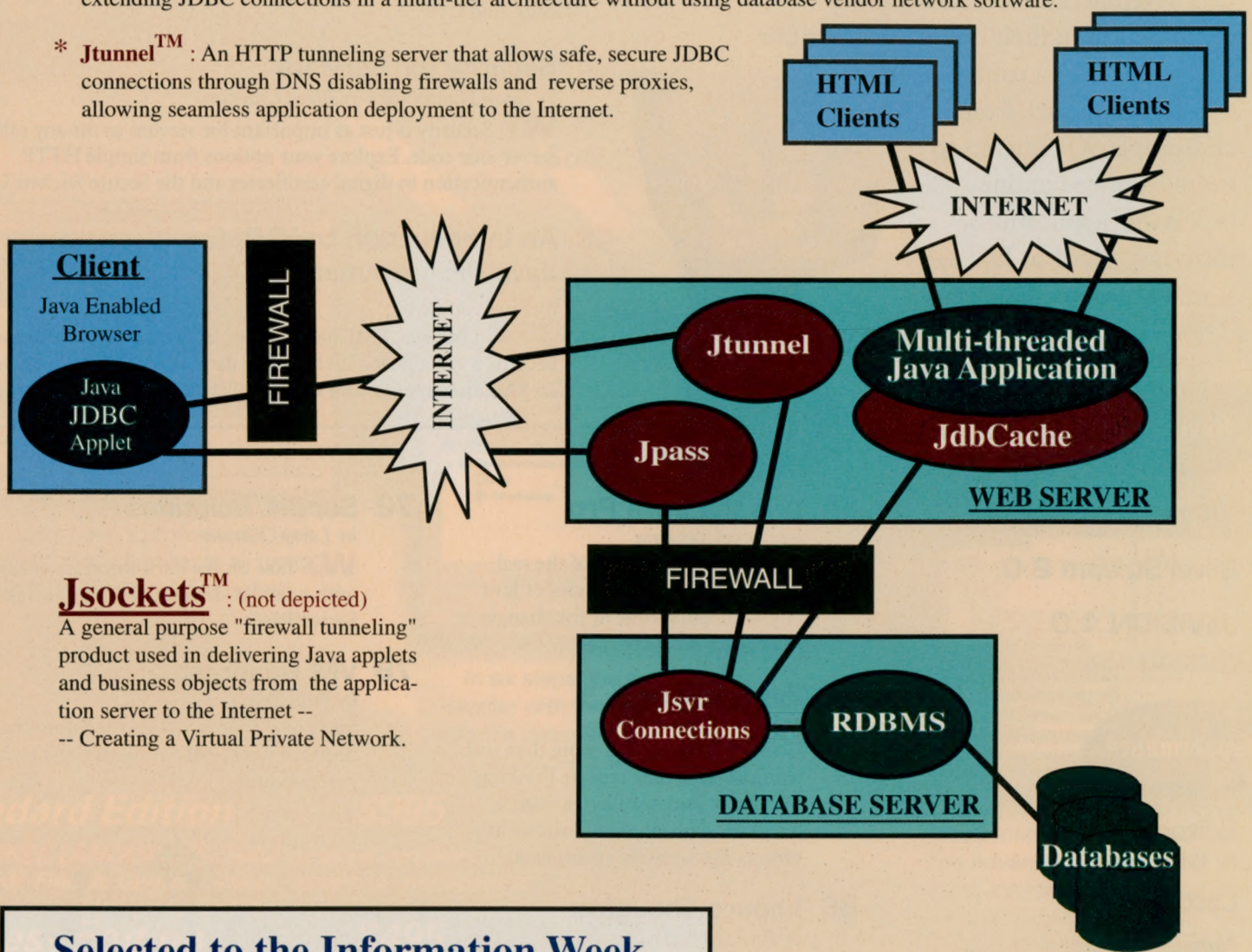
The JSQL™ 3.0 Software Suite from Caribou Lake Software

* **Jsvr™**: A robust, scalable Type 3 JDBC connection server that is highly configurable with client specified time-outs, meta-data, BLOB support, complete connection logging and advanced security features.

* **JdbCache™**: A service that pools, reuses, and serves database connections to applications. This allows a Java application to use a small number of database connections to service many requests. It's ideal for multi-threaded Java programs because the threads do not have to create and use their own JDBC connection. Includes usage statistics.

* **JPass™**: Used to "pass-through" JDBC calls from client applications to any database server via a web server, hence extending JDBC connections in a multi-tier architecture without using database vendor network software.

* **Jtunnel™**: An HTTP tunneling server that allows safe, secure JDBC connections through DNS disabling firewalls and reverse proxies, allowing seamless application deployment to the Internet.



Jsockets™: (not depicted)
A general purpose "firewall tunneling" product used in delivering Java applets and business objects from the application server to the Internet --
-- Creating a Virtual Private Network.

Selected to the Information Week Top 100 Internet Solution Series
(October 5, 1998)

Learn more about our products, free trial software, and superior professional service offerings at:

www.CaribouLake.com

CARIBOU LAKE SOFTWARE

Superior Java/ RDBMS/ Data Warehouse Products and Services

Minneapolis + Milwaukee + San Francisco

Strength in Numbers

by Tyler Sperry
Editor in Chief

By all accounts, this is a good time to be part of the Java community. The language continues to grow in popularity. What was once dismissed as a Web fad for creating dancing logos is now taken seriously as the next wave for corporate software development. While the cross-platform promise is an important element for IS managers, the productivity and reliability boosts don't hurt either.

The growth in Java usage complicates life for editors of magazines. After all, magazines succeed by delivering value to their readers. A growing, shifting readership provides a challenge in selecting topics and coverage. *Java Pro* has grown with the field by maintaining a focus on practical, "how to" Java coverage. We've been successful to the point that *Java Pro* is now the best-selling Java magazine in the market. We're gratified by your support, but we're still looking for ways to improve. Which is, not so coincidentally, where you can help.

Closing the Loop

We editors like to think—in our weaker moments, at least—that our great wisdom is sufficient to discern the articles and topics that will provide the best value to you. Given the humbling nature of experience, however, we like to hedge our bets as much as possible. Like the most-zealous graduates of a Tom Peters seminar, we eagerly use any excuse to pursue customer feedback. We survey readers constantly, looking for clues to provide a better magazine. Reader comments, from glowing compliments to X-rated nastygrams, are weighed judiciously. We blend them with cross-tabulated survey results and industry market research. Then we make our decisions and hope for the best.

Some of those decisions are easy to make, of course. We recognized servlets were an important topic, for example, long before we were able to find someone to cover the subject in a monthly column. Readers asked for more coverage of servlets, and we complied. Only recently have survey results quantified the issue and revealed that nearly half of you are using servlets in your work.

The harder decisions come when survey results lead the editors on a wild goose chase. Java's rampant popularity, for example, recently led a C/C++ magazine to plaster Java all over its cover and start running a Java column. I've got to believe that editor will be getting a ton of flame mail. The issue isn't about loyalty to one particular language; it's about the magazine's focus. We have an article in this issue, as a matter of fact, that discusses how you can support Java access to files with JNI and C++ code. You can rest assured, however, that our use of C++ will be judicious and sparing.

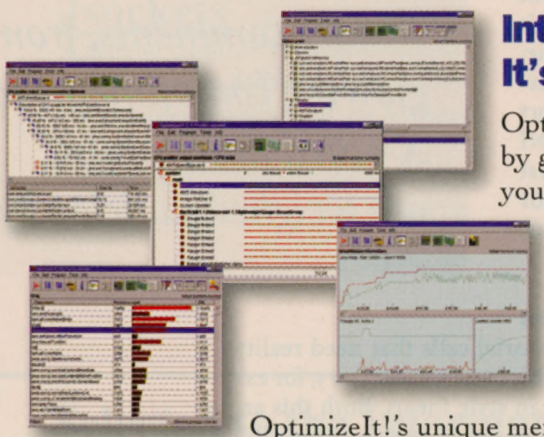
Reader comments, from glowing compliments to X-rated nastygrams, are weighed judiciously

Vote Early, Vote Often

Finally, there are some editorial calls that need reality checks from the readers. This issue's cover story, for example, marks a departure in style for us. Both this article on Java-database connections and next month's survey of XML technology are focused more on the state of the art than on specific coding techniques. Are articles of this sort as useful to you as our other articles? Let us know by ranking this issue's lineup in the reader survey on page 38 or the online version at www.java-pro.com. If you're the sort that feels constrained by check boxes, feel free to give us chapter or verse by sending your e-mail comments to java-pro@fawcette.com. We welcome all feedback. Even from those of you who want more articles on C++ and Eiffel.

Advanced Java™ profiling with OptimizeIt™!

Because you have better things to do with your time than track Java performance problems.



Introducing OptimizeIt! 3.0 Professional. It's Java Profiling Like You've Never Seen.

OptimizeIt! 3.0 Professional makes your Java profiling process easier by giving you the most comprehensive and accurate information on how your Java programs use CPU and memory. What you get is clear, real-time data and easy-to-read graphs allowing you to rapidly pinpoint performance problems such as bottlenecks, excessive object allocations or memory leaks.

OptimizeIt! is the only tool to include a CPU profiler allowing you to tune CPU time measurement to the most efficient level of detail for each test session – while maintaining a low overhead.

OptimizeIt!'s unique memory profiler also provides unequaled features such as a reference graph that automatically highlights the references actually causing memory leaks.

But don't take our word for it. Check us out at www.optimizeit.com to download your free trial today, and experience how OptimizeIt! will help you improve the overall performance of your Java programs. And just maybe get you out of the office on time.

www.optimizeit.com

Optimizeit 3.0 Professional supports JDK 1.1x and JDK 1.2

©1999 Intuitive Systems, Inc. All rights reserved. Optimizeit is a registered trademark of Intuitive Systems, Inc. All registred trademarks belong to their respective manufacturers.

JAVA Pro

Jeff Miller, *Vice President, Publishing*

Editorial
Tyler Sperry, *Editor in Chief*
John Dorsey, *Managing Editor*
Elise M. Peterson, *Associate Editor*

Contributors
Daniel Savarese, James W. Cooper
Larry O'Brien, Paul Tyma, Rick Grehan

Art & Production
Michael Hollister, *Art & Production Director*
Amy René Pelto, *Design Production Associate*
Christine Marquez, *Production Manager*
Kathy Sweeney, *Production Artist*
James Kattinge, *Senior Designer*
Kim Le, *Senior Designer*
Kimberly Bell, *Electronic Products Associate Art Director*
Robyn Wandzell, *Electronic Production Associate*
Michael Furukawa, *Associate Production Manager*

Advertising Sales
Kent Wu, *Advertising Director*
Regan Swett, *Regional Sales Manager*
Kelly Keane, *Regional Sales Manager*
Shana Deane, *Regional Sales Manager*
Monica McQueen, *Production Coordinator*
Lilly Gutierrez, *Recruitment Advertising Manager*

Circulation
Karen Koenen, *Circulation Director*
Katie Hathaway, *Associate Circulation Manager*
Patrick Bonstead, *Circulation Associate*

Marketing
Sherry Thomas-Zon, *Marketing Director*
Julianne Olmstead, *Ancillary Products Manager*
Libby Dickinson, *Marketing Communications Manager*
Margaret Horoszko, *Marketing Designer*
Kristin Fladager, *Marketing Specialist*

Conferences
Tena Carter, *General Manager*
Timothy Haight, *Editorial Director*
Ann Odonnell, *Conference Manager*
Daryl Haggard, *Conference Planner*
Janet Brandt, *Conference Planner*
Fred Perry, *Conference Planner*
Robert Scoble, *Conference Planner*
Jennifer Brucker, *Conference Planner*
Tom Donoghue, *Conference Sales Director*
Damian Zusi, *Conference Sales Manager*

Electronic Products
Matthew Carter, *Editor & Producer*
Andrew King, *Associate Producer*
Bill Poulin, *Web Developer*
Kay Keppler, *Senior Editor*
Sylvia Zele, *Associate Editor*
Diane Matcheck, *Product Reviews Editor*
Mana Tominaga, *Editorial Assistant*

Electronic Products, Advertising
Duane Dirstine, *National Sales Director*
Ben Waltzer, *Sales Support Manager*
Darrin Lord, *Regional Sales Manager*
Jane Myer, *Regional Sales Manager*
Paul Seal, *Regional Sales Manager*
Amanda Tyler, *Regional Sales Manager*

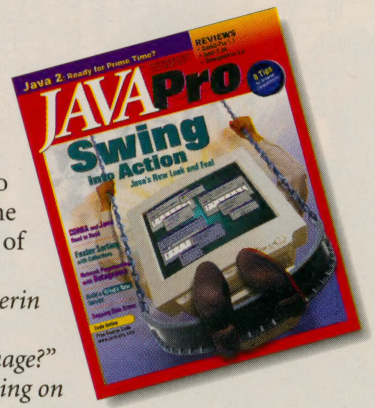
Customer Service
Andrew Nickels, *Customer Service Manager*
Michelle Grossman, *Customer Service Representative*
Chris Ortiz, *Customer Service Representative*
Melissa Ramirez, *Customer Service Representative*

Operations
Trisha Merriam, *Assistant to the Publisher*
Wendy Hartman-Carr, *Controller*
Aaron Wuele, *IS Manager*
Ha Hoang, *IS Specialist*
Raylene Leavitt, *HR/Executive Administrator*
Lorenzo Valencia, *Facilities Manager*
Sam Azzouni, *A/R Clerk*
Daniel Hernandez, *Receptionist*
Ashley Stringfellow, *Office Assistant*

Fawcette Technical Publications
James E. Fawcette, *President*
Lynne Matthes, *Group Publisher*
Paul Spyksma, *Publisher, FTP Advertising Network*
Stephen Evanczuk, *Vice President, Electronic Products*
Chris Kinsman, *Vice President, Technology*
Jeff Miller, *Vice President, Publishing*
Wilson, Sonsini, Goodrich & Rosati, *Corporate Counsel*

Letters to the Editor

Letters to Java Pro *are welcome. To be considered for publication, letters must include your name, address, and daytime phone number. Letters may be edited for form, fit, and style. Please send them to Letters to the Editor,* Java Pro, *c/o Fawcette Technical Publications, 209 Hamilton Avenue, Palo Alto, CA 94301-2500; fax them to 650-853-0230; or e-mail them to java-pro@fawcette.com.*



Has Java Proven Itself?

I'm a computer-engineering student, and I'm really interested in the plight of the Java language. Do you think that Java is here to stay? What about the criticism that Java hasn't proven itself in the area of sophisticated applications?

McCorney Severin

Editors respond: Is there a "the plight of the Java language?" Java has grown tremendously in popularity. Depending on whose numbers you trust most, there are between 300,000 and 900,000 serious developers. Its acceptance is way ahead of what C++ experienced at the same age.

Is Java here to stay? Of course. Has Java proven itself in the area of sophisticated apps? The same question was asked about C++ years ago. Although there are some areas that will probably stay C++ havens into the foreseeable future, the surveys of several research firms predict that Java will become the predominant development language for the majority of the Fortune 1000 by 2002. For performance-intensive, number-crunching research apps, people will probably still use Fortran or C++. In areas like middleware and business logic, it appears that C++ offers no serious competition to Java in the long run.

Has Java proven itself for sophisticated apps?

Network Programming

I enjoyed your article "*Network Programming in Java*" in the December 1998 issue. I noticed that you put the `socket.close()` method calls in the try clause for the various classes (such as NetSockets01). In another book I read, it suggested that these calls should be in the finally clause of the try clause. Is there a good reason for one way or the other? To my understanding, if it's in the finally clause, the socket is always closed upon exiting the try clause.

Tom Douglas

The author responds: This is an excellent point, and I agree that it would probably be better to put the call to the close() method in a finally clause. That way, it would be called even if an exception is thrown and control is passed to the catch block. As a long-time C++ programmer (C++ doesn't have a finally clause) I'm prone to forget about the availability of finally in Java.

Dick Baldwin

e-business

FREE Web Self-Service Starter Pack includes DB2® Universal Database™ trial code, tools and customer examples. www.software.ibm.com/websoftware.

IBM, DB2, DB2 Universal Database, AIX, the e-business logo and Solutions for a small planet are trademarks of International Business Machines Corporation in the United States and/or other countries. Sun, Solaris, Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States and/or other countries. Windows NT is a trademark of Microsoft Corporation in the United States and/or other countries. ©1998 IBM Corp. All rights reserved.

Lose the wait. Put service online with DB2 Universal Database,

the database designed for e-business. Fully Java™-enabled, it serves up text, graphics, sound and video to bring self-service apps to life on the Web. All with "right now" performance for thousands of simultaneous users – so everyone's first in line. On the back end, built-in connectors make it easy to consolidate data from virtually any source (including competitive databases) on any platform. Speaking of which, DB2 Universal Database itself runs natively on a world of platforms: from Windows NT® to Sun™ Solaris® and AIX®. The trial code is free at www.software.ibm.com/websoftware. Where you just might discover your company's first beneficiary of Web Self-Service is you.

Solutions for a small planet™ IBM.

HOW TO REACH US

Editorial Offices:
Fawcette Technical Publications
209 Hamilton Avenue
Palo Alto, CA 94301-2500
Phone: 650-833-7100, Fax: 650-853-0230
Web: www.java-pro.com, E-mail: java-pro@fawcette.com

Article Submission
To submit articles for publication please contact:
Tyler Sperry, Editor in Chief, *Java Pro*
E-mail: tsperry@fawcette.com
Download the Author Guidelines at
www.java-pro.com/authguid.htm

Product Reviews
To submit products for review please contact:
Diane Matcheck, Product Reviews Editor
E-mail: dmatcheck@fawcette.com

Product Announcements
Please e-mail press releases to: pressreleases@fawcette.com

Reprints and Permissions
For all *Java Pro* editorial and advertising reprints contact:
PARS International Corp.
114 East 32nd Street, Suite 503
New York, NY 10016
Phone: 212-779-4469
Fax: 212-779-4277

To quote from an article, please contact:
Monica McQueen
209 Hamilton Avenue
Palo Alto, CA 94301-2500
Phone: 650-833-7147, E-mail: mmcqueen@fawcette.com
Specify the issue date and title of the article, the portion you would like to quote, and the purpose.

Photocopy Rights:
Permission to photocopy for internal or personal use may be granted by Fawcette Technical Publications. Please contact Monica McQueen for more information.

Customer Service and Subscription Information
For subscription orders, inquiries, or address changes please contact Customer Service:
Java Pro
P.O. Box 54584
Boulder, CO 80322-4584
Phone: 303-945-5282, Fax: 650-321-3818
Foreign and Canadian orders must be payable in U.S. dollars, plus postage. The surface rate to Canada and Mexico is $52.97 per year. For all other countries the air-mail rate is $78.97 per year.

Back Issues
To order *Java Pro* back issues, call 650-833-7100 or 800-848-5523, and ask for Customer Service. Back issues cost $10. Additional postage will be charged to deliveries outside the U.S.

ONLINE RESOURCES

Registered Users
Locator+ code: **JP0399**
Type this code in the Locator+ field at the top of the *Java Pro* home page. You can find all the code in the current issue of *Java Pro* at www.java-pro.com. All the listings and associated files discussed in the articles are available free to Registered members in one zip file. To access code in previous issues, type "JP" followed by the month and year. For example, the Locator+code for the October 1998 issue is JP1098.

Premier Club Members
Members of the Premier Club of the Development Exchange (www.devx.com, $20 for six months) can get each article's listings and associated files in separate ZIP files. For example, to download the individual code file associated with "Ask the Pro," by Daniel Savarese, you would type in JP0399AP as the Locator+ code. Premier Club members can also access archives of all articles and related code ever published in *Java Pro*.

Java Zone
The Java Zone (www.java-zone.com) on the Development Exchange (www.devx.com) offers additional Java resources including product reviews, programming tips, weekly downloads, and discussion forums.

Letters to the Editor

Continued from page 7.

Richard Baldwin's Java Networking series in *Java Pro* is an excellent tool. Listing 1 (*Network Programming in Java*, December 1998) showed how to use the methods of the InetAddress class to access network address information. However, when connecting through a proxy server, I am unable to resolve the name www.austin.cc.tx.us with its IP address. I am able connect when specifying the IP address instead. Is there a possibility of connecting through a proxy server and resolving the host name?

Craig Epstein

The author responds: If I understand you correctly, you are on the back side of a proxy server and are unable to use the InetAddress class to do a reverse lookup. I don't know what your proxy server is doing. I am able to do this from my house, so it may be that the proxy server is interfering somehow. Perhaps you could ask your sys. admin. about the proxy server.

Dick Baldwin

I loved Richard Baldwin's article in the December issue of *Java Pro*. I am working on a Java e-mail application (send only), and I am perplexed because I cannot get my app to work for Web-based e-mail addresses like johndoe@zdnet-mail.com. I have tried to send to snap, hotmail, and ZDNet without success. I can send e-mail to anyone with a standard ISP account. Is there a difference?

Randy Rives

The author responds: I don't know why Web mail would be any different when receiving e-mail messages. However, if you are trying to use these services as an SMTP server, I doubt that you will find one that even has an SMTP server.

Normally an applet can only communicate with the server from which it was downloaded. That should prohibit applets from sending e-mail, but I have never tried to do it. I believe that is one of the reasons for this particular restriction.

Dick Baldwin

Richard Baldwin's Java Networking series in *Java Pro* is an excellent tool

Problems with the GIF Encoder

I downloaded the code for Claude Duguay's article (*Creating Dynamic Content with Servlets*, November 1998) which depends on the Acme JPM library. I downloaded that too, I compiled everything, and I seem to have some problems with GIF encoder (especially the StubToolkit class). Do you have a tip about what to do?

Wissam Ali-Ahmad

The author responds: The GIF encoder is limited to 256-color bitmaps. This is probably the limitation you are running into. If you use the examples I presented and extend the size of the bitmaps, you might encounter this problem because the gradients spread across a larger number of individual colors. This code is demonstrative but not designed for production use. If I were to build a commercial application based on these ideas, I would move toward JDK 1.2, which supports a JPEG encoder and buffered images, which are much more flexible.

Claude Duguay

Reviews

SilverStream 2.0
Visual J++ 6.0
JaVISION 1.0

SilverStream 2.0

Tools for Powerful Web Apps

by Luke Andrew Cassady-Dorion

Quick Facts:

SilverStream offers an environment that allows developers to graphically construct Web application functionality and flesh it out with Java code.

SilverStream Corp.
One Burlington Woods
Burlington, MA 01803
Phone: 781-238-5400
Fax: 781-238-5499
Web: www.silverstream.com
E-mail: info@silverstream.com
Price: Single-developer license,$495: single-processor deployment license, $8,500

As anyone who attempting to make money on the Internet will tell you, there's a lot of money to be made in Web applications. The folks at amazon.com, Yahoo! and eBay have had great success, and there are millions of others trying to follow in their footsteps. Many of the people making money on the Web are creating applications that build community, sell a product, or provide business functionality. Although many people are fighting for eyeballs on an overcrowded Internet, there are others attempting to make money creating the weapons used to fight the war.

Early on, Web applications were developed in scripting languages like Perl and were not terribly complicated. Companies soon realized that enough people were building Web applications to make it profitable to provide them with a collection of tools to rapidly assemble their applications. SilverStream, from SilverStream Corporation, is making quite a splash with it's recently released 2.0 version.

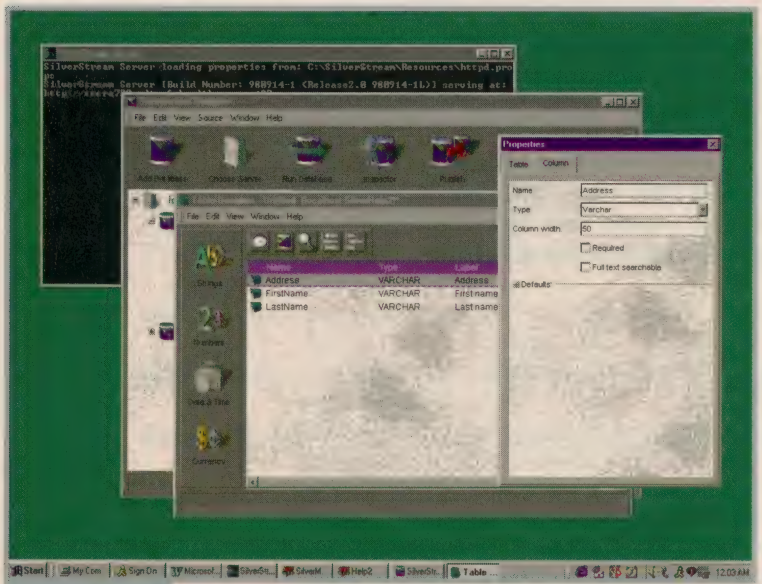
Tools used to develop Web applications target a wide variety of developers. At one end are tools that present a simple point-and-click UI and allow databases to be graphically linked to Web pages and functionality to be added. At the other end are tools that require all development to occur in a specific language such as Java or C++, and provide no graphical utilities. SilverStream offers an environment somewhere in the middle that allows developers to graphically construct functionality and flesh out that functionally using Java code.

Applications developed using SilverStream are thin-client *n*-tier, meaning the client simply gathers user information and passes it on to a middle tier. The middle-tier consists of a collection of business objects that provide functionality and communicate with the database, and the database serves to store information. The client component is developed using HTML or Java (either as an applet or application), the middle tier using the SilverStream toolset, and the database using any ODBC compatible database. The SilverStream toolset allows HTML-to-database communication.

To facilitate the development of middle-tier functionality, SilverStream provides a server application used to execute code and a designer application used to design functionality (if you desire, the designer can be replaced with your favorite IDE). The server is highly reliable and supports many of the features required by enterprise applications, including server- and session-level fail-over, load balancing, security (encryption and authentication) and easy management. Developers often spend a great deal of time writing these features themselves, their presence in SilverStream allows those them to concentrate on business functionality. The SilverStream Designer, as shown in the Figure, allows developers to write custom Java classes that communicate through the SilverStream API with the client and the database. Realizing that much of the code developed during this process can be automated, SilverStream provides tools to help build these classes. Each user interaction can be tied to any collection of business objects, exposing a path through their functionality. In addition, a business object can be setup such that it responds to an external action such as incoming e-mail or after a predetermined amount of time.

Once control flow has been transferred from a client to some business object, that object acts just like any other server-side Java object in existence. It can communicate with a database or CORBA ORB, interact with the file system or even instantiate a new network connection.

I've been developing Web applications for four years and among the myriad tools that I have used, SilverStream is a bright light I wish I had found sooner. It allows me to perform simple database interactions using the included tools and simultaneously add advanced functionality using custom Java code. ■



Designed Right The SilverStream Designer can be used to control the entire product lifecycle. Here, its ability to modify database tables is shown.

Luke Cassady-Dorion, a Sun Certified Java 1.1 Programmer, is a lead developer with Healtheon Corp. Luke has deployed many enterprise-wide solutions written entirely in Java. Luke is the author of numerous articles on distributed computing and Java, and co-author of the recently published, *Java Distributed Objects*. He can be reached at luke@luke.org.

The impossible deadline:

Deliver web access to the sales database by NOON TODAY.

No problem.

Meet the challenge with VisualCafé™ Version 3:
The fastest Java™ tool for delivering database information.

Need SQL via JDBC through HTTP and Servlets? Build it today with VisualCafé Version 3 Database Edition– the right tool to create complex, web-enabled transaction oriented database applications.

Powerful new database functionality accelerates application development:

- **Open JDBC development environment**
- **Instant Query By Example (QBE)**
- **SQL and stored procedure beans**
- **Validation and calculation rules**
- **Full support for JFC/Swing components and data model**
- **Database wizards including master detail support**

CHECK OUT OUR SPECIAL DEVELOPER OFFER

at www.visualcafe.com/vc3/partneroffer

For information on site licensing, call Symantec Corporate Sales at 1-888-822-3409

NEW! Powerful Distributed Debugging
For a preview of VisualCafé Enterprise Suite, see www.visualcafe.com/vces/walkthrough

Computer Discount Warehouse	J&R Computer World	MicroWAREHOUSE	PC Connection	PROVANTAGE	Programmer's Paradise
1-800-903-4239	1-800-221-8180	1-800-397-8508	1-800-800-4656	1-800-336-1166	1-800-445-7899

©1999 Symantec Corporation. All rights reserved. Symantec and the Symantec logo are registered trademarks of Symantec Corporation in the United States. VisualCafé is a trademark of Symantec Corporation. Oracle is a registered trademark of Oracle Corporation. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems in the U.S. and other countries. In Canada, call 1-800-365-8641. In Australia, call 02-9850-1000. In Europe, call 31-71-535-3111.

The Fastest Java Development Solution For Corporate Databases.

SYMANTEC.

Visual J++ 6.0

It's a great environment, but is it Java?

by Larry O'Brien

Quick Facts:

VJ++ makes it incredibly easy to write Win32 programs, but don't expect the same results if you want to adhere to the standard API.

Microsoft Corp.
One Microsoft Way
Redmond, WA 98052-6399
Phone: 425-882-8080
Web: http://msdn.microsoft.com/visualj/

Whatever the future holds for Microsoft's Java strategy, it is certainly worth documenting where it stands today. Microsoft professes to love "Java the language," but it views the standard Java API as a competing operating system that it intends to defeat. All that has gone before with Java is largely irrelevant to the ultimate success of the Java language and its associated API—the battle over Java is a five-to-ten year campaign that has just gotten underway. Those who use Java today are already realizing the futility of adhering to the standard Java API. The shrill voices of this trivial fraction of the development community signify nothing. They (including you, dear reader) are the Amiga owners of the programming world.

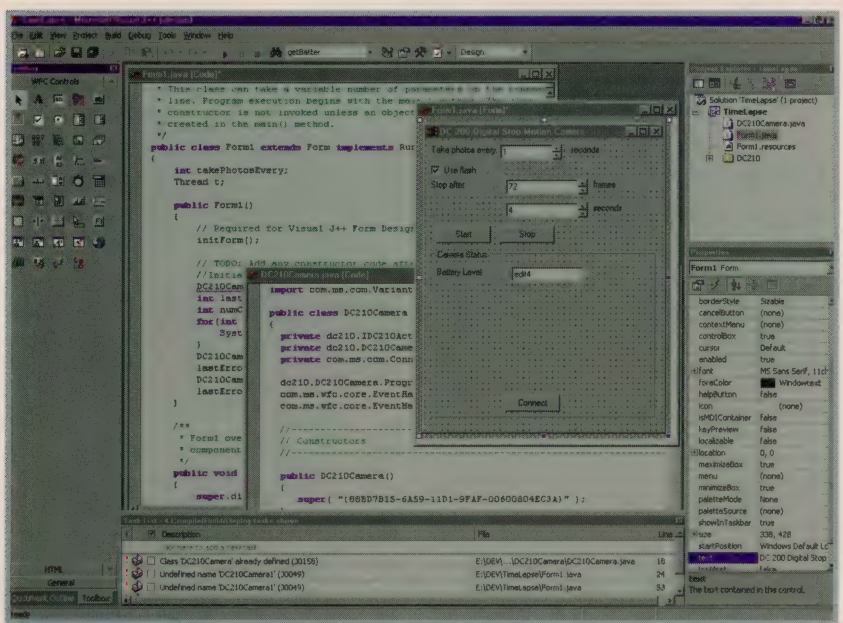
The definition of a language as syntax and semantics but not a standard API is, at best, highly debatable. Let's just say that for brevity, VJ++ is a more convenient name than "Microsoft's implementation of the Java programming language." What does VJ++ offer? A class library called Windows Foundation Classes (WFC), a programming model that fits perfectly with COM, and a beautiful development environment.

It's incredibly easy to write Win32 programs in VJ++. My first WFC program turned my Kodak DC-200 digital camera into a time-lapse video camera. After opening VJ++, two clicks brought me to the Customize Toolbox window. A click, a listbox, and another click inserted the DC210 ActiveX control into the toolbox. Click-drag-click the icon onto the form and my hard drive spun into action. In a moment, I had a subdirectory called DC210 that contained six .java files encapsulating the interface to my camera. The Properties toolbox exposed various properties of the camera, and I could attach code to various camera events with a double-click on the appropriate event. Within two hours (more than an hour of which was figuring out the camera's ActiveX control) I was creating time-lapse movies of ice cubes melting.

Other projects highlighted some of the limitations of WFC. For example, it is harder to add additional WFC components to the toolbox than it is to add ActiveX controls. But I have never seen a better excuse for a programmer to own a 21" monitor than Visual J++'s development environment. It's not just a glorified text editor, the environment treats the project and code under construction as a dynamic object even before it's compiled, and provides a multitude of screens showing different views into this object. Even at 1024×768 resolution the screen can appear cluttered. I advise 1280×1024 if possible.

It's possible to write non-Windows-specific Java apps in VJ++—just as it's possible to drive a nail with a frozen herring. If you are interested in portable Java programs or Java programs that use APIs beyond the 1.1 language level, don't use VJ++. It's not fair to say that Microsoft is attempting to seduce or confuse Java programmers, because everything about VJ++ makes it clear that users are expected to embrace Microsoft's solutions, from DirectX for putting pixels on the screen to BackOffice and Transaction Server for moving enterprise data.

Although I have reservations about Microsoft's ability to compete on the enterprise level, I have none when it comes to the desktop or the home. The only viable alternative to the Windows interface in that market is a server-provided browser-based service. If you want to create shrink-wrapped software using the Java syntax, VJ++ is a great tool. The price you pay for Windows functionality is that you must learn and use Microsoft's WFC. VJ++ is easier to program than C++ and Microsoft Foundation Classes but probably more difficult than Visual Basic. However, the expressiveness of true object orientation and the excellent fit with the COM programming model make VJ++ very attractive for shrink-wrapped Windows development. But is it Java? Does it matter?



Got 1280×1024? I have never seen a better excuse for a programmer to own a 21" monitor than Visual J++'s development environment. Even at 1024×768 resolution, the screen can appear cluttered.

Larry is an independent trainer and consultant and developed *Bruce Eckel's Hands-On Java CD*. He is the chair of the "Internet and Intranet Development" track for the Software Development conference. Reach him at larryo@altnetinc.com.

so much DATA, so little INFORMATION.

Simplify the process for understanding your data with JWAVE™'s sophisticated numerical and graphical techniques.

- **ACCESSIBILITY** universal deployment over the Internet of your data visualization application regardless of size and complexity

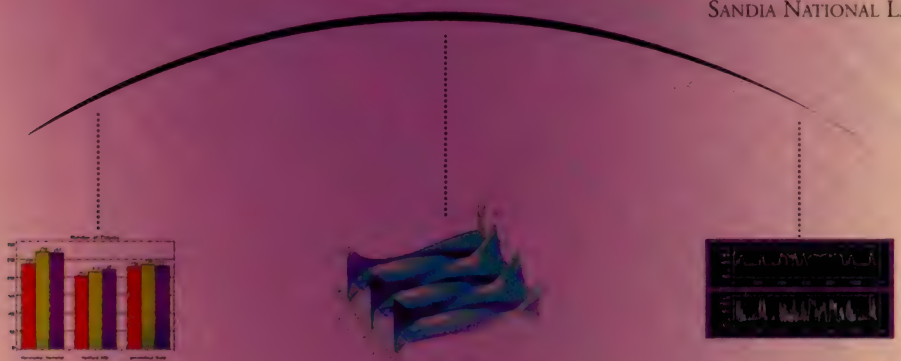
- **SCABILITY** proven repository of numerical and statistical algorithms for small and large applications in technical and business environments

- **CONSULTING** we help you build your application from project definition through development and deployment.

Understand your data. Make informed decisions. Improve your bottom line. Visit our Website at www.vni.com/javapro for more information and to register for a **FREE T-shirt**.

"One of the most important benefits we've derived from JWAVE is a huge time savings in performing data analysis. What used to take a week to accomplish, now can be done in minutes using JWAVE. We look forward to using JWAVE in many future applications."

BILL SWARTZ
MANAGER OF INFORMATION TECHNOLOGIES AND DATA MODELING
SANDIA NATIONAL LABS



Visual Numerics®
Solutions without Bounds

[WWW.VNI.COM/JAVAPRO INFO@VNI.COM 800-364-8880]

Copyright© 1999 Visual Numerics, Inc. Visual Numerics, IMSL and PV-WAVE are registered trademarks of Visual Numerics, JWAVE is a trademark of Visual Numerics, Inc. All other product and company names are trademarks or registered trademarks of their respective companies.

P/N 011999

JaVISION 1.0

Cost-effective class modeling, viewing, and spot editing

by Jacques Surveyer

Quick Facts:

JaVISION is a UML modeling tool for Java able to quickly translate Java code into color-coded class diagrams ready for viewing or editing.

Object Insight Inc.
Ann Arbor, Michigan 48104
Web: www.object-insight.com
E-mail: info@object-insight.com.
Price: $99

Now that Java has a significant presence in Web and other development projects both on the server and client, a cottage industry of design and development tools to fortify the classic IDE tools is appearing. Given the rapidly increasing value of good programmers and the need to stay on top of the fast expanding universe of Java libraries and code, CASE and Unified Modeling Language (UML) modeling tools are experiencing a revival. Object Insight has entered the arena with JaVISION.

JaVISION promises "Quick UML diagrams from Java code." But to be precise, JaVISION produces "Quick UML *class* diagrams from Java code." This is the primary, but not the only, differentiation between JaVISION, selling for $99, and Object International's Together/J or Platinum's Paradigm Plus, selling for $2,000-plus. These tools support the full range of principal UML diagramming techniques such as Use Case, Collaboration Diagrams, and Sequence Charts, while JaVISION is limited to Class diagrams. The other UML modeling tools also support OO languages like C++ or Smalltalk, with both forward and reverse engineering. However, despite its limitations, JaVISION's competitive price performance makes it a viable option as developer's Java modeling *and* spot maintenance tool of choice.

Java presents programmers with three problems: the huge number of classes and packages; the complex inheritance and interrelationship among those classes; and the need to quickly spot edit code. Okay, be honest now—who's able to keep up with all the Java class libraries, Beans, and other components? Java's richness is also its curse. And Java classes are growing like weeds. Keeping up with Java and searching/exploring intelligently the many Java packages and class libraries can be a major preoccupation of the Java developer. Not to mention the need to get a view or model how the classes, interfaces and packages fit together.

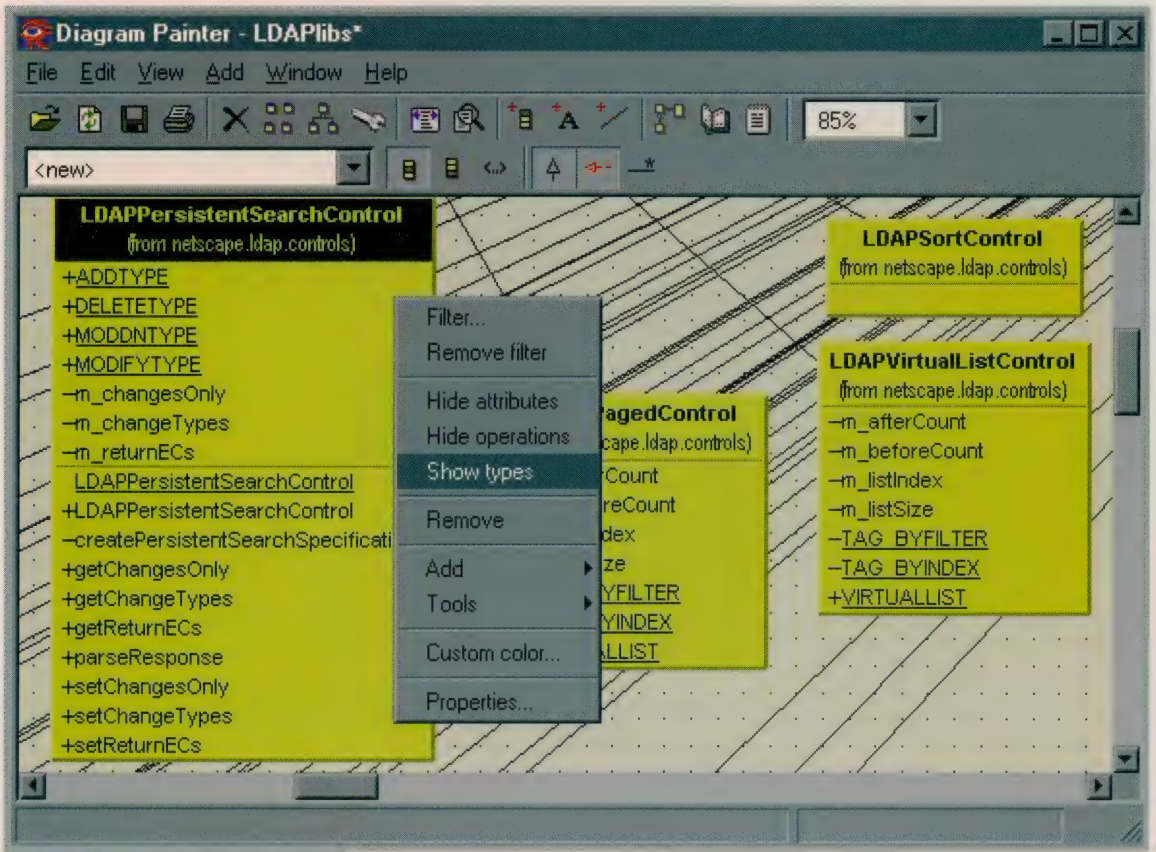


Figure 1 Filter and Hide by Type This close up view of Netscape's LDAP classes shows JaVISION's ability to filter and show or hide by type.

JaVISION as Modeling Tool

UML has taken off as the standard for object-oriented modeling. This is due in part to the convergence of analysis and design methods, and the combined efforts of Grady Booch, Ivar Jacobsen, James Rumbaugh, and others to merge competing diagramming methods and processes of OO analysis, creating a unified but extensible methodology.

Class Diagrams sit at the pinnacle of UML methodology. They are used interchangeably with Use Cases models to determine the scope and object contents of a system. Persistent classes and their relationships map very closely from Class Diagrams to classical Entity Relationship data modeling. Finally, Collaboration and Sequence Diagrams are often derived from, or directly borrow from, Class diagrams to help delineate the operations and attributes of a class. Enter JaVISION. Point JaVISION to

your classpath and Java code directory, click on the Add menu, and up pops a dialog which allows you to select which classes (from .class, .jar, and/or .java files) to diagram. Then, in five to ten seconds, dozens of classes are automatically diagrammed for you.

JaVISION provides all the conveniences of auto layout tools including automatic layout of all classes contained in all specified .java, .class, .jar files; user control of hiding/showing of class attributes, methods and parameter types; proper UML coding for `+public`, `-private`, `#protected`, `static`, `abstract` attributes, and/or operations; proper UML lines and icons for relationship between classes; and the ability to quickly zoom in and out on portions of the diagram. More importantly, JaVISION supplies a number of ways to filter, highlight, and hide portions of the Class diagram. Class displays can be filtered on any combination of Class attributes. The same is true of links for relations, inheritance, and implements can be shown/screened. When you have dozens of classes, relation links can become quite thick and hard to make sense of without color coding or filters. In Figure 1 multiple links look like tracks from an atomic particle bubble chamber. Perhaps the nicest feature is the ability to color code the classes under user control for a dozen categories (we could think of a lot more) while being able to apply dozens of pastel shades to individual classes.

However, JaVISION does have a significant weakness. It would be convenient to layout classes by name and a few attributes and not have to go into the editor each time. But JaVISION cannot forward engineer from the class diagram into code. Worse yet, some editors (including my personal favorite, Premia's CodeWright) cannot communicate back to the diagram changes that have been made after the first edit and save (it's no problem for JaVISION's own editor). Yet these same editors work perfectly well in the same capacity with designer tools like Adobe PageMill or Symantec's Visual Page. Object-Insight says that they are aware of the problem and working on it. But despite these difficulties, JaVISION addresses our Java issues effectively.

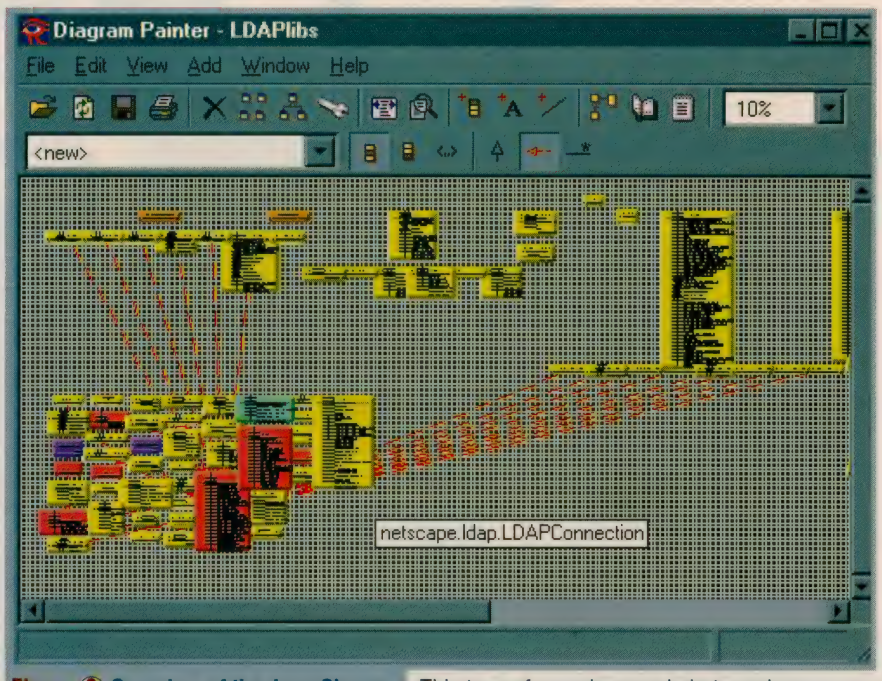


Figure 2 Overview of the Java Classes This type of overview can help to make sense of a complex Java package with color coding and selective filtering.

JaVISION as Viewing Tool

The number one problem with Java is the inability to quickly delve into Java's packages and classes and find out what they do. You can use any editor, but you won't be able to see the forest for the trees. In Figure 2 the color coding and selective filtering can help to make sense of a complex Java package. If you want to see more about a specific class just double click and up pops the class in your editor. If you want to be able to spot your specific classes quickly, just color them a "see-me" fuchsia. In short, JaVISION is well suited to viewing and making sense of the complexity of Java.

JaVISION as Spot Editor

Of course, users want to do more than view their Java code. Many Java coders would like to browse through various classes to make quick changes. Inheritance and components imply small, but critical changes to sizable blocks of code and components are divided up into modular snippets. When making changes to non-GUI Java code, most of the time is spent hunting down the right classes, adding new methods or attributes, or changing small snippets of method. In this capacity, JaVISION, with its built-in editor, can be helpful.

For example, if a client is running a Java browser-based application that takes advantage of the Lightweight Directory Access Protocol (LDAP) links in Netscape, and the new Novell Netware Directory Services for NT routines were announced, the client needs to see the impact on their connections. Instead of loading up the full Java IDE, the development staff can quickly browse through the Netscape classes, isolate which classes and methods are affected, and make changes on the spot while tentatively adding two derived classes for their specific purposes. Because JaVISION allows custom color coding the development staff can show to project and development managers the affected routines in the context of their complete sub-system. The result is a sense that crisis was averted and the program fixes and extensions were quickly deployed.

However, if the staff had one of the $2,000 UML programs available, the task would have been even easier. JaVISION would do well to add some of the better features of Paradigm Plus or Together/J to its repertoire. These two earn their keep with forward and reverse engineering capabilities for a number of languages, interfacing to popular CASE tools, and Together/J has some usability conveniences such as a thumbnail overview of the whole diagram, and active property sheets. But are these extra features worth the extra $1,900? It's up to you.

JaVISION's bargain price, rapid speed, and attractive layout of Java code into highly viewable class diagrams, make it a cost-effective tool for Java class modeling, viewing, and spot editing. ■

Jacques Surveyer is a consultant. Visit his Web site at www.inforamp.net/~jbsurv.

New Products

Platforms

Testing

Utilities

Report Generation

PersonalJava

Platforms

TowerJ 3.0

TowerJ 3.0 Java Deployment Platform allows server-side Java applications to execute with optimized native compiled performance.The TowerJ 3.0 release includes Tower's new DynaFlex runtime virtual machine, which is embedded in an application's native executable allowing dynamic updating of the application via Java bytecode, natively compiled DLL's, or shared libraries. The new release also adds a partitioning option to Tower's whole system optimizing native compiler allowing developers to create optimized native DLL's from Java bytecode classes. Call for pricing. Available Q1 1999.

Tower Technology
1501 W. Koenig Ln.
Austin, TX 78756
Phone: 512-452-9455
Fax: 512-452-1721
Web: www.towerj.com
E-mail: tower@twr.com

Centuris PKI Platform

The Centuris Public Key Infrastructure (PKI) Platform is an enterprise security solution, providing developers with a tool for creating PKI applications and integrating advanced security technology into existing e-commerce applications. Features include extensible API's, support for open standards PKCS and PKIX for interoperability with other vendors, and support for connections with virtually any database or directory, leveraging a companies investment in an existing enterprise database. Call for pricing.

Phaos Technology
200 W. 15 St., Ste. 6F
New York, NY 10011
Phone: 212-229-1450
Fax: 212-229-9704
Web: www.phaos.com

Testing

Object/Pilot

Object/Pilot is an automatic test tool that explores the distributed object space, interacts with arbitrary objects, and observes how they perform in a CORBA-based system. Graphical usability tools and automatic code generation facilitate the fast creation of test clients that can be utilized to "test-fly" servers. Object/Pilot automatically generates Java test clients that are IIOP-enabled, assuring maximum portability and interoperability with servers implemented in any programming language. Through a test client, the user can interactively apply a stimulus to any server and immediately observe the results. The ability to automatically generate test clients means that changes to a server interface or implementation can be easily propagated back in the creation of a new client. Call for pricing.

Black and White Software
1901 S. Bascom Ave., Ste. 700
Campbell, CA 95008
Phone: 408-369-7400
Fax: 408-369-7406
Web: www.blackwhite.com
E-mail: info@blackwhite.com

Utilities

InstallAnywhere 2.5 Standard Edition

InstallAnywhere is a 100% Java application that allows developers to create a single, universal installer that will run on any Java-enabled platform, automatically installing a virtual machine on the client if one isn't available. InstallAnywhere also handles all the platform-specific details. It tailors the install for the user's system, whether setting custom icons on Mac OS or creating shortcuts in the Windows Start menu. The latest version has over 150 new and improved features including JDK 1.2 support, optimization for large installations, the ability to install with or without a Java VM, and LaunchAnywhere application launchers for Windows 95, 98, NT, Mac OS, OS/2, Solaris, Linux, HP-UX, IRIX, AIX, and all other Unix platforms. $995.

Zero G Software Inc.
118 King St. # 415
San Francisco, CA 94107-1916
Phone: 415-512-7771
Fax: 415-512-7775
Web: www.zerog.com
E-mail: info@zerog.com

Metamata Suite 1.0.4b

The Metamata Suite 1.0.4b is the maintenance release of a set of integrated productivity components designed to enhance popular IDEs for developers working in the areas of static-code quality analysis, static metrics measurements, smart source code browsing, and advanced debugging.

The Metamata Suite offers the following components: Metamata Audit evaluates code for errors and style. Metamata Metrics calculates global complexity and quality metrics on code sections, and delivers object-oriented metrics. Metamata Edit is an editor that understands Java's semantic structure. Metamata Debug allows for rapid prototyping and debugging of partially written programs, and provides separate windows to control threads. Metamata Package automatically strips assertions and diagnostic code from Java programs. Metamata Parse is a follow-up tool to Sun's JavaCC parser generator for Java. Metamata Diagnostics provides the ability to extend Java programs with such diagnostic code as assertions, print statements, and so on. It can be used as a standalone program or work with Metamata Debug.

The components run in any JDK 1.1 environment. Pricing for a single-user, full suite license is $1,500; a floating license is $2,300. The Audit module can be priced separately ($395/$595); as can the Metrics module ($995/$1,495); as can the Debug module ($250/$375); the other components are free.

Metamata
2140 Peralta Blvd.,
Suite 213B
Fremont, CA 94536
Phone: 510-796-0915
Web: www.metadata.com/products/

SFS offers DocFather Pro 2.1

SFS Software announced DocFather Professional 2.1, an upgrade to its full-text, keyword-search indexing engine.

DocFather Professional contains a Java viewer, which functions as either a standalone or embeddable applet. DocFather Professional lets Web site visitors rapidly locate required documents through a customizable interface. The software can arrange for automated live updates to indexes on Web sites, and because it is a Java application, it can run on any Java-compatible system.

DocFather Professional 2.1 is available now. An Internet license for commercial Web sites costs $349; a corporate intranet license runs $990; and a non-commercial Web site license costs $99. There are also available options to use DocFather as the search engine on a CD-ROM distribution. Developers can get a demo version.

SFS Software
Allende Street 68 *
98574 Schmalkalden
Germany
Phone: 49-3683-798170
888-467-1806
Fax: 888-467-1806
Web: www.sfs-software.com/docfather/index.html

Report Generation

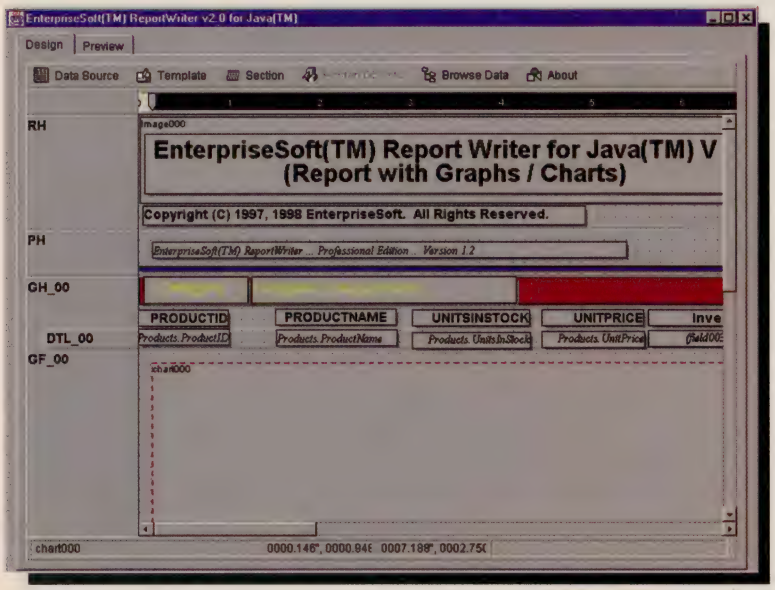
Seagate issues Crystal Reports 7

Seagate Software has released version 7 of the Java-based Crystal Reports, its ad hoc query and report-creation soft-

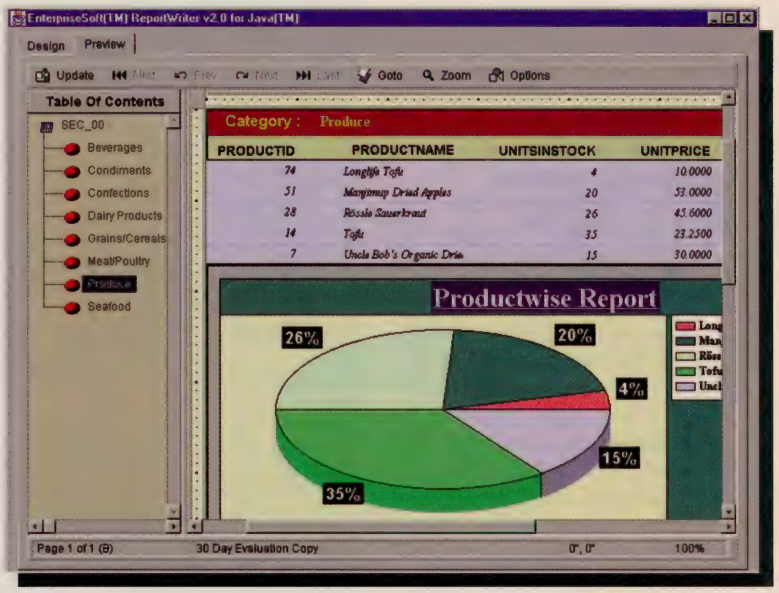
DO YOU DREAM OF PROVIDING PROFESSIONAL QUALITY OUTPUT FROM YOUR JAVA™ APPLICATIONS/APPLETS?

INTRODUCING

ENTERPRISESOFT™ REPORT WRITER FOR JAVA™ PRO V 2.0



ERW DESIGNER



ERW VIEWER

WHAT CAN ERW PRO 2.0 DO FOR YOU ?

Extract ERW is data-source independent. It can extract data from any Java application object or from a JDBC data-source. Data extraction is fast and scalable to several thousand objects.

Analyze Analyze reports with multiple sections nested groupings, charts/graphs, table of contents for report data, formulas as well as user-defined functions and query operators. The ERW report engine provides blazing fast, real-time report generation.

Format Use the familiar, easy-to-use ERW Designer window to format the report structure and use the extensive customization features to define attributes of your report data.

Output Any ERW report can be exported to the PDF, HTML or CSV file format. In addition, the reports can be viewed on screen or sent to a printer for a WYSIWYG hard copy.

THE FIRST REAL REPORTING/PRINTING SOLUTION BUILT WITH YOU, THE JAVA™ DEVELOPER, IN MIND.

- 100% Pure Java™
- Limited, free run-time re-distribution
- Small footprint (~500 KB)
- Robust, fast
- Extendable, scalable
- Easy-to-use designer and API
- Extensive docs and sample code
- Developer-based licensing - $495

Try ERW Pro 2.0 for yourself! Find out why companies such as NationsBank/Bank of America, Ford, IBM, and The Getty Museum have chosen ERW Pro 2.0 for their Java™ reporting/printing solutions! Download a 30-day free evaluation copy from our Web site today!

EnterpriseSoft™

7573 Waterford Drive, Cupertino, CA 95014 ■ PHONE 510-742-6700 ■ FAX 510-742-6800
EMAIL java@EnterpriseSoft.com ■ WEB http://www.EnterpriseSoft.com

EnterpriseSoft, the EnterpriseSoft logo, Enterprise Reports and "Enterprise-wide Business Intelligence Solutions" are trademarks of EnterpriseSoft. Java and Java related trademarks are property of Sun Microsystems, Inc. All other trademarks are property of their respective owners.

ware. Seagate Crystal Reports 7 answers and generates these queries and reports over the Web, using interactive geographic mapping, and legacy report conversion and disparate-data integration.

The new geographic-mapping features let users visualize data and drill down into maps to examine interesting areas. Cross-tabs are now customizable, and when combined with the new field highlighting feature, subtle data patterns can be highlighted so users are alerted to performance problems that before could have gone undetected.

Version 7's Document Import tool makes it easy to convert legacy text reports to the Crystal Reports format by using sophisticated pattern-recognition technology. A utility called Field Mapping facilitates the migration of existing reports between databases without having to redesign the report.

There is now added support for the latest of database technology, including SQL Server 7 and OLEDB. Integration with VB, Java, and Delphi development environments is now tighter. It comes with the Report Designer Component (which accesses features in VB 6.0), an ActiveX designer. New JavaBean and VCL components are available for Java and Delphi support, respectively.

Seagate Crystal Reports 7 Professional Edition costs $395; upgrade from previous versions for $199.

Seagate Software
920 Disc Drive
Scotts Valley, CA 95067
Phone: 408-438-6550
Fax: 408-438-7612
Web: www.seagatesoftware.com/seagateinfo/

PersonalJava

PersonalJava emulation environment 3.0

The PersonalJava emulation environment helps developers verify that the PersonalJava applets and applications you develop using the JDK will run in the PersonalJava application environment. There are four versions of the PersonalJava emulation environment available for use with specific deployment-platform targets: "Win32/Windows" for a Windows look-and-feel; "Win32/Touchable" for a Win32-platform Touchable look-and-feel; "Solaris/Motif" fora Solaris-based Motif look-and-feel; and "Solaris/Touchable" for a Solaris-based Touchable look-and-feel. All four versions support applets and applications written for either version 1.1 or version 3.0 of the PersonalJava application environment.

Sun Microsystems Inc.
JavaSoftware
901 San Antonio Road
Palo Alto, CA 94303
Phone: 800-786-7638
Web: http://java.sun.com/products/personaljava/pj-emulation.html

Kalos Espresso v3.0,

Espial Group has released Kalos Espresso v3.0, their third-generation Personal Java GUI toolkit for consumer and industrial electronic devices. Espial Group describes Kalos Espresso, as the "Swing" for embedded Java. It provides developers with a high performance, rich user interface framework for PersonalJava. With a memory footprint of less than 175K, it is optimized for minimal-resource environments

Kalos Espresso can be used to develop applications for Java-based devices such as PDAs, smart phones, set-top boxes, hand-held, mobile and other Internet devices. Using Kalos Espresso, developers can dynamically adapt their apps to the look and feel of their host device. It offers an advanced collection of widgets that include buttons, checkboxes, radio buttons, lists, tables, trees, word-wrapped editors designed to be flexible and make efficient use of limited screen real estate Kalos Espresso also offers threadsafe management without the performance overhead of traditional synchronization

Kalos Espresso can be used with Kalos Architect v1.5; a visual development environment. Call for pricing.

Espial Group Inc.
Olympic Tower East
645 Fifth Avenue
New York NY 10022
Phone: 1.888.4.Espial
Fax: 613-226.6913
Web: www.espial-group.com

TechTIP

The Purpose of the Marker Interface

One of the "clean" features of the Java programming language is that it mandates a separation between interfaces (pure behavior) and classes (state and behavior). Interfaces are used in Java to specify the behavior of derived classes.

Often you will come across interfaces in Java that have no behavior. In other words, they are just empty interface definitions. These are known as marker interfaces. Some examples of marker interfaces in the Java API include:

```
java.lang.Cloneable
java.io.Serializable
java.util.EventListener
```

Marker interfaces are also called "tag" interfaces since they tag all the derived classes into a category based on their purpose. For example, all classes that implement the Cloneable interface can be cloned (i.e., the clone() method can be called on them). The Java compiler checks to make sure that if the clone() method is called on a class and the class implements the Cloneable interface. For example, consider the following call to the clone() method on an object o:

```
SomeObject o = new SomeObject();
SomeObject ref = (SomeObject)(o.clone());
```

If the class `SomeObject` does not implement the interface Cloneable (and Cloneable is not implemented by any of the superclasses from which `SomeObject` inherits), the compiler will mark this line as an error. This is because the `clone()` method may only be called by objects of type "Cloneable." Hence, even though Cloneable is an empty interface, it serves an important purpose.

Risks of Using Reflection

Reflection plays an essential role in adding dynamism to Java. Using reflection, an application can dynamically (i.e., at run time) determine the state and behavior of any class. Reflection allows the developer to create objects that can construct new class instances and new arrays, access and modify fields of objects and classes, invoke methods on objects and classes, and access and modify the elements of arrays.

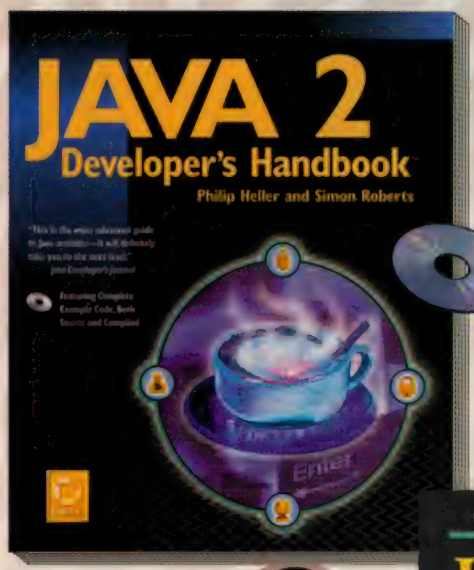
However, using the features provided by the reflection mechanism in Java requires careful consideration. The Reflection API should be used only as needed and after taking into account the costs associated with its usage. Reflection method calls have substantial performance overhead. Reflection-related code lends to overall complexity in the application. Return values from reflection method calls almost always have to be cast to the right type, making the code type-unsafe and less efficient. Errors in method invocation occur at run time instead of being detected during compilation. As a result, applications are more likely to contain bugs.

Reflection should be used only when no other simpler object-oriented language mechanisms can be used to accomplish the same task.

—Ajit Sagar

JAVA™ BOOKS FROM SYBEX®

THE MOST RELIABLE, COMPREHENSIVE JAVA BOOKS AVAILABLE



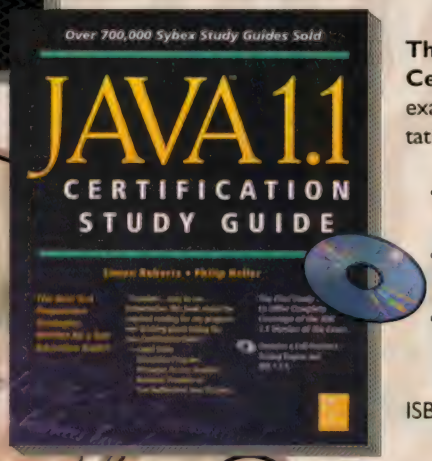
Preview a sample chapter!

The *Java 2 Developer's Handbook* is the most advanced Java development book available. Recipient of the *Java Developer's Journal* "Technical Excellence Award," this book was declared "the most advanced guide to Java available—it will definitely take you to the next level." Written by two experienced Java teachers, Philip Heller and Simon Roberts, *Java 2 Developer's Handbook* picks up where other books leave off, addressing complex issues in depth and exploring the frontiers of Java programming.

- Filled with previously undocumented tips and techniques for JDK 2.
- Geared toward the real-world development needs of experienced Java programmers.
- Build more powerful GUI with JFC Swing components and 2D API.
- Learn how Java 2 helps you program for speed and performance.

Preview a sample chapter at http://www.sybex.com/catalog/sc/Java2DH.html.

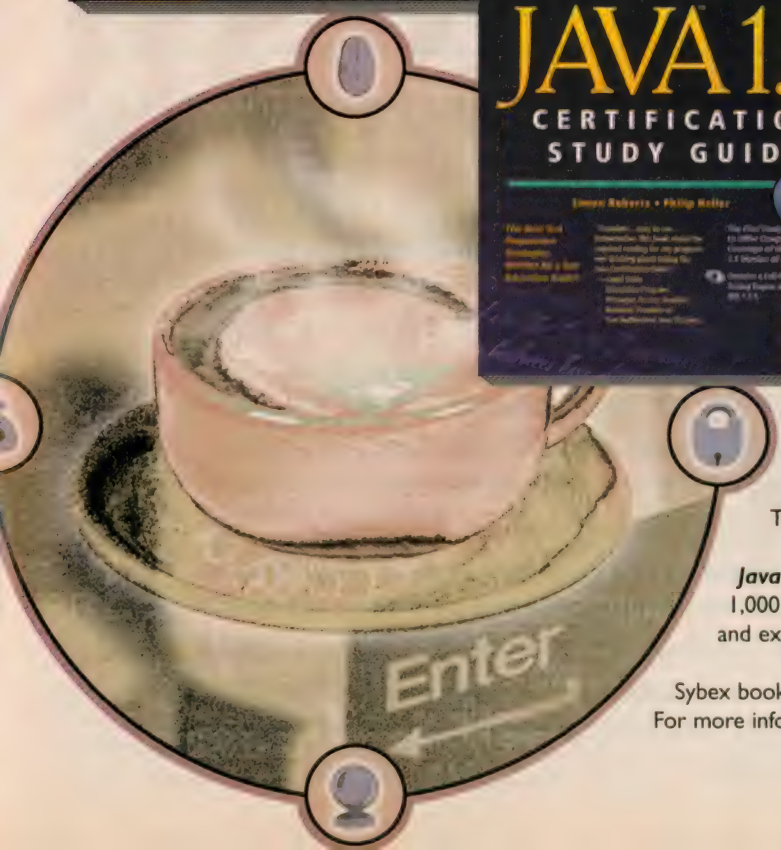
ISBN: 0-7821-2179-9 $59.99 U.S.



The Ultimate Authority for Sun Microsystems' Programmer's Certification Test! Simon Roberts, the lead author, actually developed the exam for Sun. He was the key player in the concept, strategy, and implementation of the test guidelines and he wrote the exam questions.

- Every single objective is covered—reinforced with sample questions and annotated answers.
- The book has a stamp of approval from the Sun Education Team—the people responsible for writing the exam.
- CD contains a full-featured test engine with questions modeled on the real exam.

ISBN: 0-7821-2069-5 $44.99 U.S.



OTHER JAVA BOOKS AVAILABLE FROM SYBEX:

Mastering Java 2 ISBN: 0-7821-2180-2 $49.99 U.S.
The most thorough treatment of the latest version of the Java Development Kit.

Java 2 Complete ISBN: 0-7821-2468-2 $19.99 U.S.
1,000 pages of Java information for $19.99! This Java resource includes tutorials, references, and extensive JavaScript coverage.

Sybex books are available at your favorite book seller and software retailer.
For more information about all Sybex titles, visit us online at **www.sybex.com**.

SYBEX®

Works With
JDK 1.x, JDBC 1.x, Sybase or other Java-enabled database

Java in the Database

Get the most from your database by extending it with Java

by Ken North

Java has increasingly become the tool of choice for extending database servers from Oracle, IBM, Sybase, and Informix. Since 1996 I've measured Java's database progress by polling developers attending my Java database seminars. In early 1996 many developers were asking how Java Database Connectivity (JDBC) compared with Open Database Connectivity (ODBC). Later that year developers were wondering if Java could handle porting a 10,000 seat client/server SQL application to an intranet architecture in 90 days. By the summer of 1997, more developers were writing applications and server-side code than applets. 1998 saw an increased interest in Java in the database. Throughout the entire period, 90–95 percent of those audiences were using one of the five best-selling DBMSs: Oracle, Microsoft SQL Server, Sybase, Informix, or IBM DB2.

Ken North consults, writes, and teaches Expert Series seminars. He is the author of *Windows Multi-DBMS Programming*, John Wiley & Sons, 1995 and *Database Magic with Ken North*, Prentice Hall PTR, 1998. Reach him at kennorth@interserv.com or visit his Web site at http://ourworld.compuserve.com/homepages/Ken_North/.

DBMS Vendors Embrace Java

Java's success has prompted Oracle, Microsoft, Sybase, Informix, and IBM to develop JDBC drivers for their databases. Oracle, Sybase, Informix, IBM, and Cloudscape are in a race to produce an SQL platform that best serves the needs of Java computing. However, SQL database vendors are not alone in their support for Java and JDBC. A variety of middleware vendors offer JDBC drivers and JDBC servers. For example, CocoBase, from Thought Inc., is a middleware product that maps the relationships between objects and relational databases. It enables developers to create Java stored procedures for object databases such as Versant. Middleware products such as Caribou Lake JSQL, Intersolv SequeLink, OpenLink Server, and Simba Express provide server-side JDBC, but they execute in a middle-tier server, not in the database.

Oracle, Sybase, IBM, Informix, and Cloudscape support Java database extensions. IBM DB2 Universal Database (UDB) 5.0 was the first major server product to embed the Java VM. DB2 UDB was followed by three small-footprint, mobile-computing DBMSs: Oracle Lite, Sybase Adaptive Server Anywhere (ASA), and Cloudscape JBMS.

The next generation is comprised of top-of-the-line servers from Oracle, Sybase, and Informix. By the end of 1999, Sybase Adaptive Server Enterprise, Oracle 8i, and Informix Dynamic Server will be Java enabled. They will also include an embedded Java VM, execute Java stored procedures, and act as a container for Enterprise JavaBeans. Microsoft SQL Server 7.0 shipped with a JDBC driver, but it did not include support for Java in the database. However, Microsoft has said a future release of SQL Server will support "language-independent" stored procedures.

For more information, log on to the following Web sites:

- ***Caribou Lake Software:*** www.CaribouLake.com
- ***Cloudscape JBMS:*** www.cloudscape.com
- ***IBM DB2 Universal Database:*** www.software.ibm.com/data/db2
- ***Informix Dynamic Server:*** www.informix.com
- ***Intersolv SequeLink:*** www.intersolv.com
- ***JDBC Drivers and Middleware Vendors:*** http://ourworld.compuserve.com/homepages/Ken_North/JDBCvend.htm
- ***Microsoft SQL Server:*** www.microsoft.com/sql
- ***OpenLink Server:*** www.openlinksw.com
- ***Oracle 8i:*** www.oracle.com
- ***SimbaExpress:*** www.simbatech.com
- ***Sybase Adaptive Server:*** www.sybase.com
- ***Thought Inc. CocoBase:*** www.thoughtinc.com —*K.N.*

Although my surveys weren't scientific, it appears that the first wave of Java database developers were primarily concerned with traditional types, transactions, and performance. The second wave were doing more server-side programming, but many were still using traditional SQL tables. I expect that the next wave will include Java developers using more complex types and multidimensional databases. This is because SQL vendors are moving to object-relational DBMSs that support database extensions (or plug-ins). I will discuss Java's role in extending databases, but to get started I'll provide some background information about extensible server architectures, active databases, and how we are using databases today.

OLTP, OLAP, OLCP—What's the difference?

Databases have a wide variety of uses and there are different requirements for online transaction processing (OLTP), online analytical processing (OLAP), and online complex processing (OLCP). Understanding these differences is important in selecting an appropriate architecture and DBMS, designing databases, and choosing an API. OLTP, OLCP, and OLAP databases each answer different questions. For example, to find orders for greater than 1,000 widgets you can use an OLTP query against an operational data store that contains current information. Amazon.com uses SQL queries against OLTP databases. OLTP programs place a premium on short-lived queries to support many concurrent database users. Transactions typically involve SQL tables containing rows of numbers and characters. If you want to find a picture containing a sunset image you can use an OLCP query against a database containing images. OLCP applications use databases that contain more complex types, such as audio, images, video, text, and spatial data. Finally, to get a list of the top-ten selling books in California for each month in the period 1995–1997, you can use an OLAP query against a multidimensional data source containing historical information. OLAP involves decision support queries against multidimensional data sets. OLAP programs typically sustain few users, doing long-running queries.

Java is a viable technology for OLTP, OLAP, or OLCP data access. However, the database design and data access APIs are often different for OLTP, OLAP, and OLCP. The JDBC API and SQL are well-established for OLTP, but developers creating decision support queries are likely to use an OLAP server with a different API. The major OLAP vendors endorse OLE DB for OLAP, and multidimensional expressions (MDX) that map to SQL. For OLCP, Java developers often use Java bindings for object DBMSs such as Gemstone and Versant. The OLCP picture is changing because SQL servers are becoming object-relational and JDBC 2.0 provides better support for object-relational databases.

Java is a viable technology for OLTP, OLAP, or OLCP data access

SQL and Active Databases

To grasp the importance of the new Java capabilities, you need to understand some SQL DBMS concepts. SQL databases store logic and enforce rules about data. They are active databases because they can trigger events and block operations that compromise data integrity. Some SQL servers can be extended by programming and installing a dynamic link library (DLL), a shared library, or Java classes. The extensions are a part of the database, and they execute at the server. Java can also contribute to the database's role as a logic store and rule enforcer, and Java can add user-defined functions (UDFs), stored procedures, types, and behaviors to a database. Finally, Java extensions are accessible to any SQL client. The DBMS parses SQL statements and invokes Java classes, whether the client is Visual Basic, a middle-tier server, Netscape Communicator, or Internet Explorer.

Description	Package Name
Server-side JDBC	(com.sybase.*)
Primitive Java types	(java.lang.integer)
JDBC	(java.sql.*)
Sun packages	(sun.*)
Utility package	(java.util.*)
Sybase classes	(Sybase.sql.*)

Table 1 Java-Powered Sybase These packages are required to created a Sybase Java-enabled database.

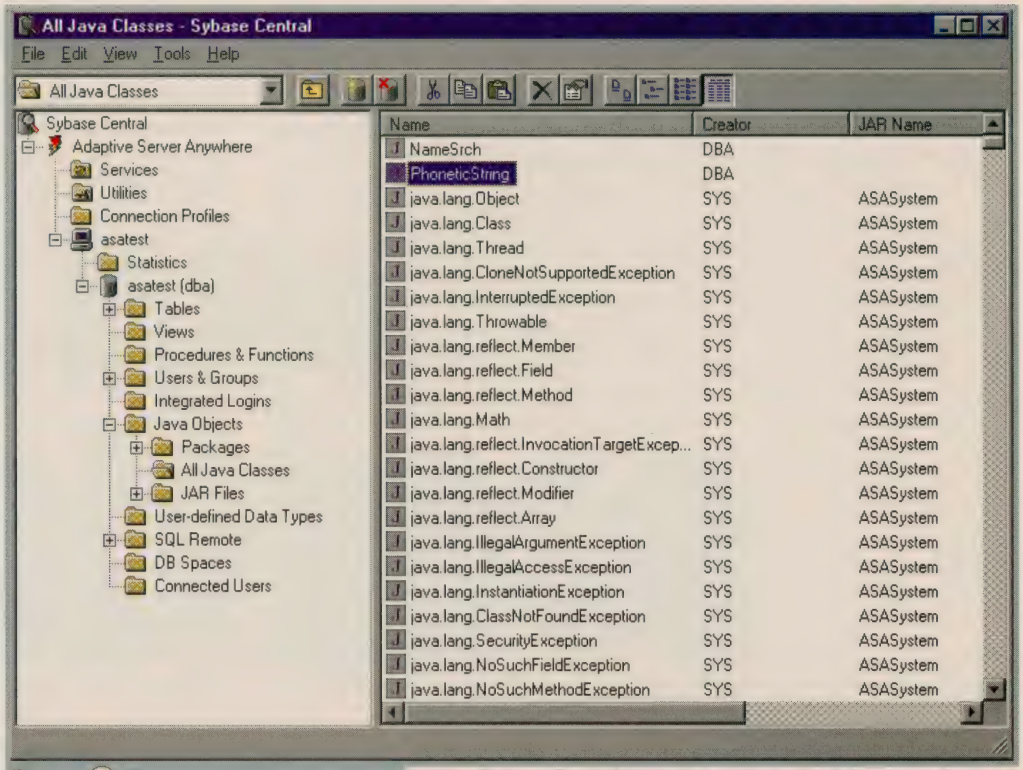


Figure 1 The Java-Enabled Database In the Sybase Central view of a Java-enabled database the database manager treats Java classes as schema objects. (Reprinted by permission from *Database Magic with Ken North*, Prentice Hall.)

Although many organizations are running legacy databases, SQL enjoys widespread support among Java database developers. Part of SQL's appeal is that it is an international standard that is updated every few years. Most SQL DBMSs conform to the SQL-92 standard (see http://web.ansi.org/public/std_info.html for more information), but many vendors use proprietary SQL dialects for stored procedure programming. Oracle provides PL/SQL, Informix offers Stored Procedure Language (SPL), and Microsoft and Sybase support Transact-SQL (T-SQL) stored procedures. IBM DB2 Universal Database uses a different model, supporting external stored procedures written in programming languages such as Rexx, COBOL, C, and Java.

PL/SQL, T-SQL, and SPL are block-structured, procedural languages in their own right, but they aren't portable from one DBMS to another. On the other hand, Java represents the most portable programming solution available today. Java's portability is important to IBM, Informix, Oracle, and Sybase because they offer SQL servers for different operating systems. Processing statistical functions, string manipulation, and time–series analysis is more efficient when coded with a programming language like Java instead of extended SQL dialects.

Understanding the Architecture

There are several architectures used for Java programs that query a database with SQL. You can write an applet that downloads to the client when someone uses a Java-enabled browser. You can write Java programs, application and servlets, that do not execute in a browser context. Of course, before executing an application or servlet, you must pre-install the Java VM and runtime classes on the machine where your program will run.

To the database server, a client-side program, such as an applet or application, and a server-side program, such as a servlet. are all SQL clients. Some SQL servers are extensible with C/C++ and other languages, but Java is likely to become the standard for server-side programming because it's flexible and portable. In addition to applets, servlets, and middle-tier servers, Java developers can create classes to install in a database. Java-enabled browsers include the Java virtual machine and runtime classes. SQL servers that execute Java classes also include the Java VM and runtime. A database that embeds a VM is called a "Java-enabled" database. For example, a Sybase Java-enabled database includes JDBC classes, core classes for types, and other packages (see Table 1). Figure 1 illustrates a Sybase Central view of a Java-enabled database. The database manager views Java classes as schema objects just like tables, columns, views, triggers, and stored procedures.

Embedding Java in Database Servers

Java-enabled SQL servers offer different capabilities, and not every vendor uses the same approach to embedding Java in the DBMS. Cloudscape is an embeddable DBMS that uses the Java VM of the host application or operating system. DB2 UDB uses different versions of the Java VM for different operating systems. Oracle and Sybase use their own Java VM across all operating system platforms. Sybase uses the same Java runtime and VM for both of its server products (Adaptive Server Anywhere and Adaptive Server Enterprise).

There are several benefits to using a VM and runtime classes optimized for use in a database server. Java in the database doesn't require classes, such as AWT, to support a GUI. A Java VM tailored to database server requirements should provide optimized garbage collection and thread management. Responsiveness for OLTP requires certain types of queries to be given priority for resources such as memory and threads. You also want the server to inhibit Java garbage collection when it is doing high-priority queries.

SQLJ and JDBC

SQLJ is a proposed standard for embedded SQL for Java. A SQLJ preprocessor reads a source file

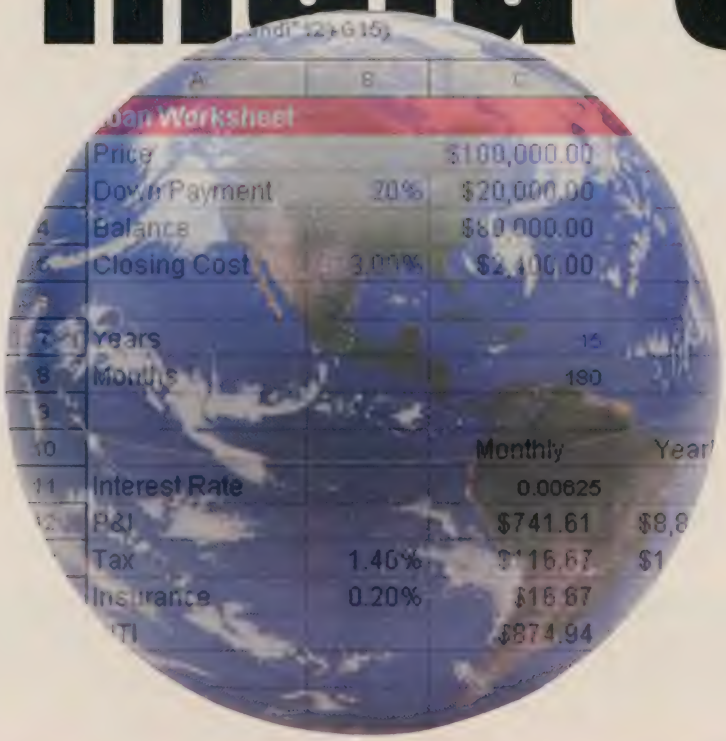
Why Java?

There are a number of reasons why Java is an important tool for extending SQL databases:

- Java offers a single paradigm for programming all tiers of a multitier application.
- Developers can use it to program clients and servers.
- Java supports application partitioning and thin clients by moving logic into middle tiers or into the database.
- Some databases will soon be Enterprise JavaBeans containers, making Java necessary to extend them.
- Java moves SQL databases closer to the object-relational model by supporting custom types and behaviors.
- Proprietary SQL dialects inhibit writing portable UDFs and stored procedures.

—*K.N.*

Formula One™



Technology for delivering Spreadsheets Everywhere.™

Java.
Web pages.
Multiple platforms.
Web-based applications.
Server-side functionality.
Spreadsheets.

Formula One for Java ties them all together, empowering you to deliver spreadsheets everywhere.

With Formula One for Java, Java developers are immediately armed with a powerful spreadsheet JavaBean for use in environments such as Visual Café and JBuilder.

Use it on a server to calculate and format data. Use it in Internet or intranet applications. Connect it to databases. Or read and write Excel files.

The same fast calculations and robust spreadsheet functionality is also available to Webmasters, who can use it to add live spreadsheets to their HTML pages with as few as one line of code.

Even end users on any desktop can use it as a standalone application.

Thousands of businesses such as State Street Bank and Simon & Schuster trust Formula One in their applications.

You, too, can leverage superior spreadsheet functionality and Java in your next application.

Tie them together.

Deliver spreadsheets everywhere.

Use Formula One for Java, the Internet Spreadsheet™.

Formula One™ Professional

- ✓ Formula One for Java® 5.5.1
  - ✓ Application, applet, and JavaBean
  - ✓ Read/write Excel files
  - ✓ 100% Pure Java
- ✓ Formula One ActiveX™ 6.0
  - ✓ Read/write Excel files
  - ✓ First Impression ActiveX 6.0 charts
- ✓ Fast calculations
- ✓ Excel formula and function syntax
- ✓ Workbook designer
- ✓ Robust API
- ✓ Database connectivity
- ✓ Small footprint and downloads
- ✓ Server and client side functionality
- ✓ Write spreadsheets to HTML
- ✓ Per user and server/CPU pricing

Tidestone TECHNOLOGIES™

INFORMATION:
800-922-9665

FREE FULL-FEATURED TRIALS:
www.tidestone.com/ssheets

JDJ EDITOR'S CHOICE AWARD

Copyright © 1999 Tidestone Technologies, Inc. All rights reserved. Tidestone, the Tidestone logo, Formula One, Spreadsheets Everywhere, Internet Spreadsheet, and First Impression are trademarks of Tidestone Technologies, Inc. Java, 100% Pure Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. All other trademarks are property of their respective owners. Specifications subject to change without notice.

containing SQLJ code (.sqlj file) and emits a Java (.java) source file. SQLJ simplifies programming SQL databases and enables developers to use precompiled SQL. JDBC uses dynamic SQL, but SQLJ enables developers to use static SQL. SQLJ is important for developers writing Java stored procedures. Oracle included SQLJ with Oracle 8, IBM started shipping SQLJ with DB2 UDB version 5.2, and Informix and Sybase will release SQLJ in the first quarter of 1999.

JDBC consists of Java classes that abstract the process of writing SQL programs. JDBC provides connection objects, result set objects, metadata, and three types of statement objects. The first statement type supports ad hoc queries, the second supports prepared queries, and the third supports stored procedures. JDBC also includes classes that wrap SQL data types such as timestamps. (For more information about JDBC, consult *JDBC Database Access with Java,* Graham Hamilton, et al., Addison Wesley, 1997.) When Rick Cattell, Graham Hamilton, and Mark Hapner of JavaSoft specified JDBC 1.*x*, they were consistent with ODBC and SQL-92. However, SQL DBMSs are evolving into object-relational DBMSs, and JDBC 2.0 introduces SQL3 features that support the new object-relational databases.

JDBC Drivers and Connections

A server-side JDBC program appears similar to client-side code, but there are differences. For client programming, JDBC developers use several types of drivers: the JDBC-ODBC bridge (type 1), a driver that maps JDBC to a native API (type 2), a pure Java driver for a middleware server that supports JDBC clients (type 3), and a pure Java driver that connects to a database server (type 4). Type 1 and 2 drivers use native libraries on the client. Type 3 and 4 drivers download to the client and do not use native libraries on the client. Middleware vendors offer type 3 drivers that support connections to disparate DBMSs. The major DBMS vendors offer type 4 drivers, and some offer more than one type. To connect to DB2, for example, an applet can load IBM's type 4 driver (`COM.ibm.db2.jdbc.net .DB2Driver`). An application can load `COM.ibm.db2.jdbc.app .DB2Driver`, a type 2 driver that uses the Client Application Enabler. The client uses a different connection string for applets and applications.

For a Java-enabled database, the JDBC driver architecture is different. A server-side program connects to a JDBC driver installed with the Java infrastructure on the database server. If you write a class for Sybase Adaptive Server, your code uses an internal JDBC driver instead of Sybase jConnect. Clients and server-side programs use different connection strings with JDBC's `getConnection()` method. An applet connects using the database URL and passes arguments such as `userid`, `password`, `host name`, and `port number`. This applet code connects to a DB2 database, where the `url`, `userid`, and `password` strings have been preset:

```
Connection con = getConnection(url,
  userid, password);
```

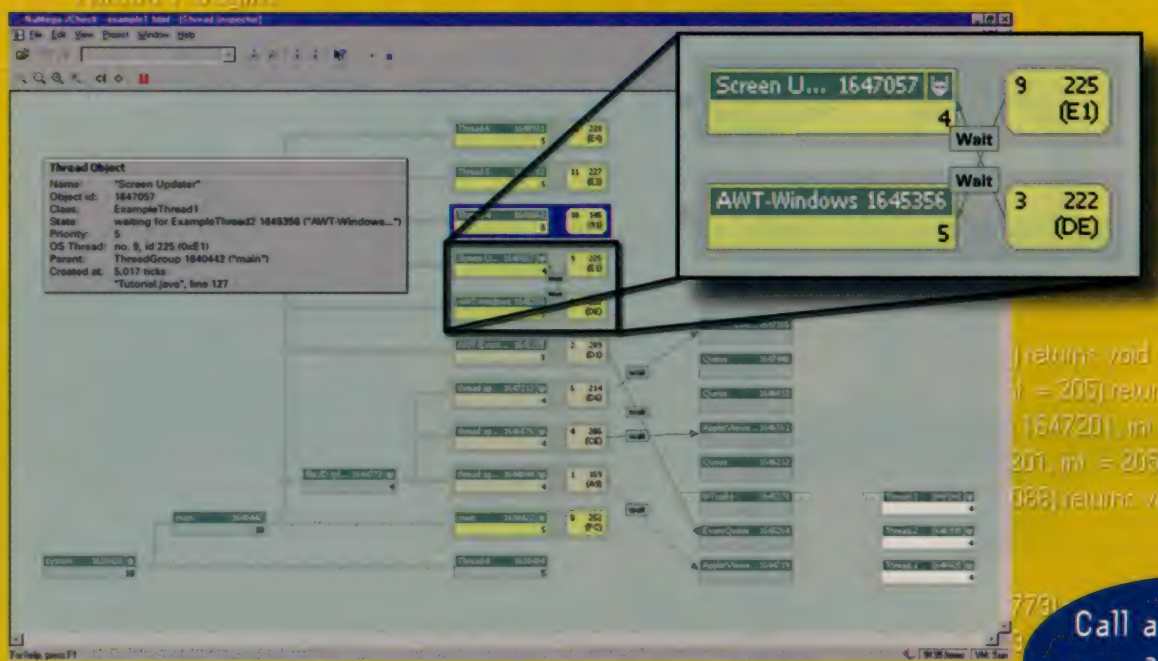
A JDBC program that installs in the database uses a null connection string:

```
Connection con = getConnection();
```

Listing 1 Server-Side Java in the Database SrchCont installs in a Sybase ASA6 database. It searches the contact table of asademo, the Adaptive Server Anywhere sample database. You pass SrchCont a "sounds-like" value for a city name. It does a SOUNDEX search of the cities in the contact table, and returns the last name and first name of the contact in that city.

```
/*********************************************************/
/* Perform SOUNDEX search on city.                       */
/* Install in Java-enabled database.                     */
/*********************************************************/

  // import JDBC classes
import java.sql.*;

public class SrchCont {

    // data members
    public String sqlQry;

    private Connection conn;
    private Statement stmt;
    private ResultSet rs;
    private boolean isMore;

    public SrchCont(String cname ) throws SQLException {

        // use internal JDBC driver connection
        conn = DriverManager.getConnection("");
        stmt= conn.createStatement();

        sqlQry = "select last_name, first_name ";
        sqlQry = sqlQry + "from contact ";
        sqlQry = sqlQry + "where SOUNDEX(city) = ";
        sqlQry = sqlQry + "SOUNDEX(\'" + cname + "\')";

        // echo class and query to Adaptive Server
        // status window
        String msg1 = "Class SrchCont: ";
        msg1 = msg1 + sqlQry;
        System.out.println (msg1);

        // execute the query
        rs = stmt.executeQuery(sqlQry);

        // position the cursor at the first row of
        // the ResultSet
        isMore = rs.next();
    }

    public String getName() throws SQLException {

        String fName = "";
        String lName = "";
        String eName = "";

        if ( isMore )
            {
            lName = rs.getString(1);
            fName = rs.getString(2);
            eName = lName + ", " + fName;
            }

        isMore = rs.next();

        if ( eName == "" )
            {
            return null;
            }
            else
            {
            return eName;
            }
    }

    public boolean setName( String cname )
        throws SQLException {

        sqlQry = "select last_name, first_name ";
        sqlQry = sqlQry + "from contact ";
        sqlQry = sqlQry + "where SOUNDEX(city) = ";
        sqlQry = sqlQry + "SOUNDEX(\'" + cname + "\')";

        // execute the query
        rs = stmt.executeQuery( sqlQry );

        isMore = rs.next();
        return isMore;
    }

    public boolean more()
        {
        return isMore;
        }
}
```

DEADLOCK!



Call and request a FREE 14 day license

JCheck™ Exposes Java™ Thread Problems Instantly

Deadlock. Thread Leak. Starvation. Thrashing... Just a few of the problems you'll never have to chase down manually again – if you use NuMega JCheck™, the visual thread and event analyzer for Java. JCheck builds a thread-based graphical model of your program at runtime and provides rich, detailed information on thread states, events, synchronization objects, exceptions, and errors. JCheck expands your runtime visibility in a new dimension, so you will waste less time debugging, and have more time for productive development.

Take your Java development to the next level. Get NuMega JCheck today!

JCheck is also part of NuMega DevPartner™ for Java
Essential tools for Java development

NuMega JCheck™ Visual Thread and Event Analyzer

NuMega TrueTime™ Automatic Performance Analysis and Optimization

NuMega TrueCoverage™ Automatic Code Coverage Analysis

1-888-686-3438

www.numega.com

COMPUWARE
People and software for business applications™

30-Day Money Back Guarantee

Copyright © 1998 Compuware Corporation. All rights reserved. All trademarks and registered trademarks are the property of their respective owners.

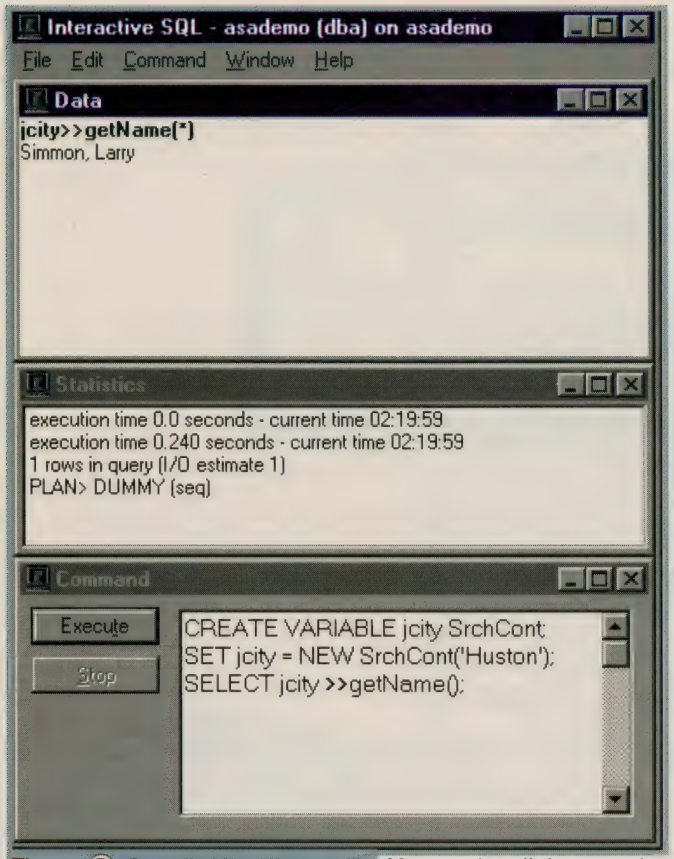


Figure 2 Sound's Like "Houston" After you install the example SrchCont class from Listing 1 in a Sybase Adaptive Server Anywhere database, you can do phonetic searches of city names. The window at the bottom shows commands for searching, and the window at the top shows search results.

User-Defined Functions

A SQL DBMS provides built-in or intrinsic functions for performing operations such as calculations, conversions, aggregation, and string processing. Many SQL DBMSs also support user-defined functions (UDF) in SQL statements. In some instances you create the UDF with SQL statements, and in others you install a library or Java class that implements the UDF. If you have an employee table that stores annual salary for some employees and hourly pay for others, and you want to execute a query that uses weekly salary as a filter, you can create a UDF that returns weekly salary as a function of classification (salaried or hourly). The query might look like this code, where `weekly` is the name of the UDF:

```
SELECT surname, first_name FROM employee
WHERE ((weekly(salary, classification)) >
  400.);
```

For a DBMS, such as DB2 or Adaptive Server, to recognize the UDF you must register it using a SQL `CREATE FUNCTION` statement. When you execute a `CREATE FUNCTION` the database manager includes the function in the database catalog (metadata). IBM DB2 enables you to use different programming languages, including Java, to create UDFs. When you register a function with `CREATE FUNCTION` you can specify execution options such as parameter-calling conventions and whether the function executes in the server address space.

Stored Procedures

A stored procedure is part of the logic of your application that executes on a database server. Depending on the DBMS, you code procedures in a SQL dialect or a programming language. Although the International Standards Organization (ISO) released a standard for procedures (or persistent stored modules) in 1996, database companies coalesced around Java as a stored procedure language.

Stored procedures use distributed processing. When a SQL client invokes a stored procedure, control is transferred to the server. The server executes the procedure and optionally returns data to the client. Procedures can return output parameters, and sometimes query result sets. Stored procedures minimize network round-trips between the client and server and they improve performance by decreasing network latency. Database administrators also use procedures to provide security. For example, you can give a user the right to execute a procedure that processes an order without giving the user rights to view or change order tables.

When you program procedures using Java, you must register them with the DBMS using the `CREATE PROCEDURE` statement. The syntax varies from one DBMS to another, but `CREATE PROCEDURE` typically identifies the type and size of input (IN) parameters, input/output (INOUT) parameters, and output (OUT) parameters. The syntax for executing a procedure varies, but there is a standard syntax you can use with JDBC drivers, ODBC drivers, and OLE DB providers. For example, to return a list of tennis players with a rating of 50, you could use this statement, where 50 is an input parameter:

```
{CALL bycatg(50)}
```

Sybase Adaptive Server Anywhere (ASA) lets you use stored procedure calls to the main method of classes installed in Java-enabled database. Therefore, you could execute this Sybase query to invoke the main method of a `ClearTotals` class:

```
CALL ClearTotals.main()
```

Example Class

Listing 1 is an example Java class (`SrchCont`) that installs in a Sybase ASA database. `SrchCont` searches the contact table of the example database (`asademo.db`) that installs with ASA 6.0. After you compile `SrchCont` and install it in `asademo`, you can use it to do a phonetic search of the city column by passing `SrchCont` a "sounds-like" value for a city name. It does a `SOUNDEX` search of the cities in the contact table, and returns the last name and first name of the person residing in that city. To install `SrchCont.class` in the `asademo` database, use a command such as this:

```
INSTALL JAVA NEW FROM FILE
  'D:\\Examples\\Java\\SrchCont.class'
  ;
```

Be sure to change the path information to match your path. After installing `SrchCont`, you can run `SELECT` queries. For example, use these statements to find someone living in a city that sounds like Houston:

```
CREATE VARIABLE jcity SrchCont;
SET jcity = NEW SrchCont('Houston');
SELECT jcity >>getName();
```

When you execute these statements, your results should look like Figure 2.

The Crystal Ball

Industry pundits seem to think that Java is good for nothing more than creating applets. However, problems such as download delays are irrelevant when it comes to server-side Java and Java in the database. Major database vendors are committed to Java as a solution for server extensibility. Even VM performance issues will not cause them to change course, as shown by IBM's native compilation of Java stored procedures on its OS/390 mainframes. If a peek into the crystal ball shows database projects in your future, then it's time to add Java to your skill set.

Code Online

All the code in this article is available for free at www.java-pro.com. Use the Locator+ codes shown below to download. Free online registration is required. See the Online Resources box in "Letters to the Editor" for more information.

Java in the Database
Registered Locator+: **JP0399**
Premier Locator+: **JP0399KN**

GRAB IT WHILE IT'S HOT
OFFER EXPIRES SOON!!

LIMITED TIME
INTRODUCTORY
OFFER $995

IS THIS YOUR JAVA TOOL'S IDEA OF SUPPORTING TEAM DEVELOPMENT?

Then you obviously don't use PARTS® for JAVA™ Professional, a tool set designed to fully support teams building distributed applications.

PARTS was created by developers with over 10 years experience designing object-oriented development tools for team programming. Our experience will make a difference in your team's productivity.

PARTS will coordinate your team's efforts with integrated support for PVCS, making it simple to track changes, compare versions, and manage your development projects. Plus, our unique tools help you understand how your code interacts with other pieces of the system, enabling you to clearly see the impact of changes.

PARTS has everything you need to be productive:

- Visual programming—just drag, drop and deploy
- JDBC support for fast, easy database applications
- JFC/Swing for powerful interfaces
- CORBA for distributed applications
- JavaBeans for quick application assembly
- OracleLite, OrbixWeb 3.0, PVCS, jKit/Grid— All integrated for a combined value of $1700!

And all backed by our world class training and consulting program chosen by both Oracle® and Sun®.

Call now for more information on how PARTS will make your team more productive—
1.800.973.4777

PARTS® FOR JAVA™
PROFESSIONAL

ObjectShare®

16811 Hale Ave., Suite A, Irvine, CA 92606 www.objectshare.com

PARTS is a registered trademark of ObjectShare, Inc. All other referenced products are trademarks and/or registered trademarks of their respective manufacturers.

BE PRODUCTIVE.

Works With

JDK 1.1.x
Microsoft Java SDK 2.0/3.x
Microsoft JVM build 3155 (or later)
Windows NT

Using Memory-Mapped Files in Java

Share data between apps written in Java as well as other languages

by Tom Guinther

In a recent article on Java and Win32 Interprocess Communication ("Interprocess Communication with Java", January 1999) I discussed the design and implementation of Java classes that exposed Win32 Interprocess Communication (IPC) objects to your Java applications. Memory-mapped files (or File Mappings) are the most powerful form of IPC available on the Windows platforms, and a majority of Win32 IPC mechanisms are built using them. This article will provide an in-depth exploration of the power and flexibility that memory-mapped files provide to the Windows developer. It also includes a JNI implementation and sample code to expose the most useful features of memory-mapped files to Java developers.

Memory-mapped files are a core feature of any Win32 operating system and are used extensively by OS code. A discussion of the primary features and benefits of memory-mapped files is in order, but it will require the use of some OS lingo. Bear with me as I attempt to translate the gibberish into a concise explanation of why the feature or benefit is important to the typical Java developer.

Perhaps the most important feature of memory-mapped files is the ability of the OS to use them to make the most efficient use of system memory (especially when the file is being shared by more than one application). Memory-mapped files allow you to map an entire file into the virtual address space of a process with physical storage provided by the disk image of the file, taking advantage of OS algorithms that automatically manage committing and decommitting physical memory. The entire file appears to be in application memory just as if you have opened the file, allocated memory, and read the entire file contents into memory. In reality, the OS reserves enough space in virtual memory for the entire file, but it doesn't actually read it into physical memory. When you access a portion of the file, if the portion hasn't been previously read (or has since been discarded) the OS transparently commits physical memory and loads the requested portion of the file. You can also use the normal caching and virtual memory commit or decommit strategies provided by the OS.

Tom is a freelance software developer specializing in systems software and internals. Reach him at tomg@nh.ultranet.com.

And for the Java Developer

One of the features supplied by memory-mapped files of interest to Java developers is the ability to share data between multiple applications, regardless of what language they are written in. Memory-mapped files allow two or more applications to share data with each other (IPC). When creating a File Mapping object you can specify an object name that can be used by cooperating applications to share information. In some cases you will share an actual disk file that contains proprietary information, or you can create a File Mapping object that is backed by the OS paging file. If you use space in the system-paging file, you can avoid the need to create a dummy file or open an actual disk file. When all instances of the File Mapping object are closed, the system will reclaim the paging file space, and any information stored there is lost (but you can always save important information to an actual disk file).

Memory-mapped files also create regions of virtual memory that are backed by the operating-system paging file, and provide data coherency when data is being shared between multiple applications. In general, views created from the same File Mapping object are guaranteed to be coherent because the OS is using the same physical memory across all applications. Therefore, a change made by one application is instantly reflected in all sharing applications. Of course, coherency isn't the same, and it doesn't solve synchronization issues, so you still need to coordinate access to the shared data between the different applications (using a Mutex, for instance.)

Additional Benefits

There are a few additional features provided by memory-mapped files. For instance, by using copy-on-write, changes

Listing 1 Simple I/O App This version of MemoryMappedFileApp uses three of the four higher level memory-mapped file I/O classes. It performs basic file input and output, and includes an example of accessing data in a shared memory buffer served from a separate process.

```
public class MemoryMappedFile {
// open file if it exists, else fail
public static final int FILE_OPEN_EXISTING      = 0 ;
// create a new file, fail if it exists
public static final int FILE_CREATE_NEW         = 1 ;
// open an existing file or create a new one
public static final int FILE_OPEN_ALWAYS        = 2 ;
// read-only access
public static final int FILE_MAP_READ           = 4;
// read-write access
public static final int FILE_MAP_READWRITE      = 8;
// Copy-on-write (read-write access)
public static final int FILE_MAP_COPYONWRITE    = 16 ;

// handle to underlying C++ implementation
protected int fileMapData = 0 ;

// information about the file mapping
protected String fileMapName = null ;
protected String fileMapShareName = null ;

// the JNI implementation that talks Win32 to the OS
private native int fileMapCreate(String fileName,
  int access, int mapSize, String shareName)
  throws IOException;
private native int fileMapOpen(String shareName, int access)
   throws IOException;
private native void fileMapClose(int fileMapData)
  throws IOException ;
private native void fileMapDestroy(int fileMapData)
throws IOException ;
private native void fileMapView(int fileMapData, int offset,
  int length, int access) throws IOException ;

// open a file mapping by its "share"
// name w/ specified access
public static native MemoryMappedFile Open(String shareName,
  int access) throws IOException ;

// create a file (mapping) read-only
public MemoryMappedFile(String fileName)
  throws IOException ;

// create a memory mapped file with the
// specified access and size to grow file
// so that new information can be added
public MemoryMappedFile(String fileName, int access)
  throws IOException ;

// create a memory mapped with the specified access
// and size to grow the memory mapped file
// so that new information can be added
public MemoryMappedFile(String fileName, int access,
  int growSize) throws IOException ;

// create a "shared" file (mapping) with the
// specified access and size to grow the memory
// mapped file so that new information can be added
public MemoryMappedFile(String fileName, int access,
  int growSize, String shareName) throws IOException ;

// map a specific portion of the file into view
public void MapViewOfFile(int offset, int length,
  int access) throws IOException ;

private native int fileMapRead(
  int fileMapData, int offset) throws IOException;
private native int fileMapReadBytes(
  int fileMapData, int offset, byte b[],
  int off, int len) throws IOException;

private native int fileMapWrite(int fileMapData,
  int offset, int b) throws IOException;
private native int fileMapWriteBytes(
  int fileMapData, int offset, byte b[],
  int off, int len) throws IOException;

private native int fileMapAvailable(int fileMapData,
  int offset) throws IOException;

  // standard InputStream and OutputStream methods
public int read(int offset) throws IOException ;
public int readBytes(int offset, byte b[],
  int arrayOfs, int len) throws IOException;
public void write(int offset, int b)
  throws IOException ;
public int writeBytes(int offset, byte b[],
  int arrayOfs, int len) throws IOException;
public int available(int offset) throws IOException ;

  // close all associated objects
  // (file, file-mapping, and view) ;
  public void close() throws IOException;

  // destructor
  public void finalize() throws IOException;
}
```

made by one application do not effect other applications sharing the file, allowing you to map sub-views so that very large files (>4 GB) can be processed easily and efficiently. Memory-mapped files also let you create sparsely committed areas of virtual memory that are automatically reflected to other applications that are sharing the file mapping.

Copy-on-write is a powerful mechanism that allows an application to modify the contents of a memory-mapped file or shared memory without effecting the data seen by other applications. A virtual game map that describes the layout of the game universe is a good example. The game map contains the initial location of items and inhabitants and anything else needed by the game universe. When the game starts, the universe file is mapped using copy-on-write so that as the game progresses changes are reflected in the current application but the original game universe is unaltered. The OS implements copy-on-write by trapping initial writes to pages in the file and cloning a page prior to it being modified (so that the original is intact). The OS uses space reserved in the paging file to buffer any modified data. Files that are larger than the 4 GB of OS address space can't be mapped in their entirety, so the OS provides the ability to map only a portion of the file at a time.

Creating Memory-Mapped Files

Memory-mapped files are the most powerful form of IPC available on the Windows platform

Memory-mapped files are Win32 specific, so the Java class implementation relies heavily on the Java Native Interface (JNI) to communicate between the Java Virtual Machine (JVM) and the Win32 OS. Before discussing the details of the JNI implementation it is important to understand the steps necessary to create and use memory-mapped files via the Win32 API. Without this understanding it would be difficult to understand the overall Java class design.

There are three basic steps to creating a memory-mapped file. First, open the underlying file that you wish to map. If you wish to map a simple text file, or a binary file with spread-

Listing ② MemoryMappedFileApp This is an abridged version of the MemoryMappedFile class. MemoryMappedFile is the base class and acts as a bridge between Java and the JNI implementation.

```
class MemoryMappedFileApp {
public static void main(String args[]) {
    try {
      int c ;

      String output = "This is a test of file I/O using" +
        " memory-mapped files" ;
      char rawOutput[] = output.toCharArray() ;

      // Create a new file using a File Mapping
      MemoryMappedOutputStream newMappedFile =
        new MemoryMappedOutputStream(
          "Test",FILE_CREATE_NEW) ;

      // write the string to the file
      for(int x=0 ; x < rawOutput.length ; x++) {
        newMappedFile.write(rawOutput[x]) ;
      }
      newMappedFile.close() ;  // important

      // Open a memory-mapped input stream,
      // read-only (by default)
      MemoryMappedInputStream mappedFile =
        new MemoryMappedInputStream("Test");

      // display the file to the console
      while((c=mappedFile.read()) != -1) {
        System.out.print((char)c) ;
      }
      mappedFile.close() ;

      // open a memory mapped share that contains binary info
      // NOTE: This data is served by another process
      MemoryMappedShare memShare =
        new MemoryMappedShare("JavaMemShare");
      DataInputStream memShareData =
        new DataInputStream(memShare.input()) ;

      try {
        do {
          c=memShareData.readInt() ;
          System.out.println(c) ;
        } while(true) ;
      } catch(EOFException e) {
      }

      memShare.close() ;

    } catch(Exception e) {
    }
  }
}
```

sheet data, use the CreateFile API to gain access to its contents. (This step is considered optional when you want create a shared memory region, in which case you don't actually need a file as the OS will provide space in the system paging file.) Second, create a File Mapping object to describe logical mapping of the file. The File Mapping object describes the file being mapped, the size of the mapping, the protection attributes, and the share name (if any). When you create the File Mapping object the OS reserves a portion of virtual memory for the files content and, if necessary, space in the system-paging file. Finally, create a view of the file data within the region of memory reserved by the File Mapping object. The view can encompass the entire file or only a portion of the file. If you need to process files larger than 512 MB, you can create a view of a smaller portion of the file, process that portion of the data, and create a new view that maps a different portion of the file.

Designing a set of classes to expose the functionality of memory-mapped files is challenging. Memory-mapped files are so flexible that nearly anyone who uses them has a customized implementation. In this situation it is most useful to implement a powerful set of base classes that expose a broad spectrum of functionality. Although the design may end up being a hodgepodge of functionality, and might not meet your normal object-oriented design standards, you will be well served by this approach. First, it is simple to create a set of easy-to-use classes (on top of the base classes) that provide access to the most commonly used features of memory-mapped files. Second, the implementation will be centralized into a small set of classes. With the appropriate comments and/or documentation, other developers should be able to uncover the necessary steps to implement more advanced or specialized functionality.

The code accompanying this article has one base class, `MemoryMappedFile`, that contains a variety of methods. To simplify, I opted not to include certain esoteric functionality, but it can be easily added if you feel comfortable modifying the underlying C++/JNI implementation. Four basic Java classes are built around the functionality of the `MemoryMappedFile` class. Each of the classes is designed to provide a basic subset of the overall memory-mapped services and directly derive from standard classes provided by the java.io package.

FileOutputStream lets you append new data to any previously existing data in the file

The first two classes, `MemoryMappedInputStream` and `MemoryMappedOutputStream` provide the ability to read from and write to (respectively) a stream that is back by a memory-mapped file. They provide the fundamental building blocks to leverage other Java classes that process information via InputStream and OutputStream. For example, the DataInputStream and DataOutputStream classes allow Java code to read and write primitive types. This is important for anyone who wants to share non-ASCII data between one or more processes. The third class, `MemoryMappedIOStream`, is based on the concepts embodied in the standard Java class, RandomAccessFile. `MemoryMappedIOStream` provides methods for reading and writing to a memory-mapped file, as well as giving the developer control over the exact point from which information is read or written. One significant difference from the RandomAccessFile class is that because `MemoryMappedIOStream` is using standard Java stream classes, it doesn't need to implement the DataInput and DataOutput interfaces directly. Instead, you can create a DataInputStream or DataOutputStream using the underlying streams provided by the `MemoryMappedIOStream`. The result is equivalent functionality without reinventing the wheel. The fourth and final class, `MemoryMappedShare`, is designed to simplify the process of creating or opening a shared memory buffer served by another process. It is a lightweight wrapper around class `MemoryMappedIOStream`.

Implementing JNI

Before we discuss the details of the Java and C++ JNI implementation, review the sample application in Listing 1. It shows a simple Java application that uses three of the four higher-level memory-mapped file I/O classes. It performs basic file input and output, and includes an example of accessing data in a shared memory buffer served from a separate process. Because the memory-mapped I/O classes are built upon the existing Java I/O framework, we could easily replace the `MemoryMappedInput-`

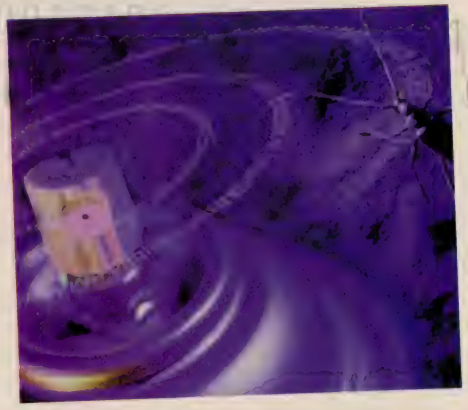
Become the Master of your domain.

Introducing the Objectsoft family of java tools

DIFFMASTER
Works with any file-based Version Control system showing you the differences in your code down to the method level.

BREWMASTER
100% Java! Featuring class browser, dynamic syntax parsing, compile management, code query capabilities, Cafe integration, J++ integration and much more.

COMMENTMASTER
Makes your documentation job easier by reading your source code and correcting the comments to conform to the javadoc standard.



DEBUGMASTER
Offers a complete debugger environment for JDK or MS JView environments.

[All tools downloadable at our Web Site]

www.objsoft.com

Objectsoft, Inc.

1-888-OBJSOFT

Stream and MemoryMappedOuputStream with other stream types, such as FileInputStream and FileOutputStream.

Listing 2 contains an abridged version of the MemoryMappedFile class. MemoryMappedFile is the base class and acts as a bridge between Java and the JNI implementation. Many of the methods are native implementations, and each one has a C++ entry point that conforms to the JNI specification. Parameters to these methods are translated (as necessary) by the JNI layer and forwarded to an underlying C++ class that provides the interface to the Win32 APIs for manipulating memory-mapped files. For performance considerations, the basic data items associated with the memory-mapped file are managed by the C++ implementation. The MemoryMappedFile class contains a handle to an instance of the C++ memory-mapped object that is passed as the first parameter to each native call. By using this type of architecture we avoid making a large number of calls back into the JVM, which could result in less than optimal performance. A second reason for using this type of architecture is that the C++ implementation of memory-mapped files is highly independent of JNI. This approach gives you a lot of flexibility for future implementations and is similar to how you would design an interface to existing legacy C++ code where it might not be feasible to integrate JNI into the existing code base.

Designing a set of classes to expose the functionality of memory-mapped files is challenging

There are a few differences between the design and implementation of class MemoryMappedFile and other stream types (such as file or pipes.) The most important difference allows the Java developer to take advantage of all the features of memory-mapped files by providing the ability to specify how the files are opened and/or created. You can specify the protection attributes for the underlying file, File Mapping object, and view(s) onto the File Mapping object. For example, the standard Java I/O classes generally don't provide precise control over how files are opened/created, provide a way to set share permissions, or directly control the read/write attributes of a particular stream. This is due in part to the fact that under most circumstances these characteristics are implicitly specified by which stream class you use to process data. For instance, creating a FileOutputStream implies that you want to be able to write data, therefore the file is opened with write permissions. Not only does the particular class imply the permissions, in some cases the constructor chosen to create the instance specifies additional functionality. The FileOutputStream contains an overloaded constructor that allows you to specify whether you want to append new data to any previously existing data in the file. The Java classes for memory-mapped files take advantage of this type of constructor overloading to simplify the obvious cases. To provide precise control over the various permissions and other options, the base class, MemoryMappedFile, uses a set of binary values that can be combined (via the | operator) into one set of flags. The options supported are listed in Table 1 along with a brief description of each flag's purpose.

Setting Permissions

Usually, it is reasonable to use the default values assumed by the class and constructor, but there are specific situations where you might override the defaults. For example, you might open or create a shared memory buffer (which uses FILE_MAP_READWRITE) and later decide to create a view onto the data that is read only. This type of functionality is possible because the Win32 API provides a very fine granularity when specifying permissions on files, File Mapping objects, and views onto File Mapping objects. The second major difference is that a memory-mapped file, unlike a standard file, is essentially random access memory, so there is no associated file pointer. In order to implement the semantics of class InputStream and OutputStream, it is necessary to allow an offset into the file to be specified when reading or writing data. MemoryMappedInputStream and MemoryMappedOutputStream maintain a current file pointer so that when you read or write, information is retrieved or stored in the correct place. When using these classes, the file position is transparent to your code (as shown in Listing 1); however, as you might expect, you can use the mark() and reset() methods to directly control the pseudo file pointer.

Additional Resources

In addition to the simple apps demonstrated here, Premier Club members can download other memory-map samples.

The project and source code for the implementation of memory-mapped file support is provided as a Microsoft Developer Studio workspace, but is easily ported to other environments. The workspace includes projects for the JNI DLL that contains the C++ implementation, the JAVAPro.Win32Ipc package (available online) that contains the Java implementation for each class in the package (including memory-mapped files), as well as projects to build and exercise the sample code. For convenience the binary files are also included. ■

Code Online

All the code in this article is available for free at www.java-pro.com. Premier Club members can also access additional codes. Use the Locator+ codes shown below to download the code. Free online registration is required. See the Online Resources box in "Letters to the Editor" for more information.

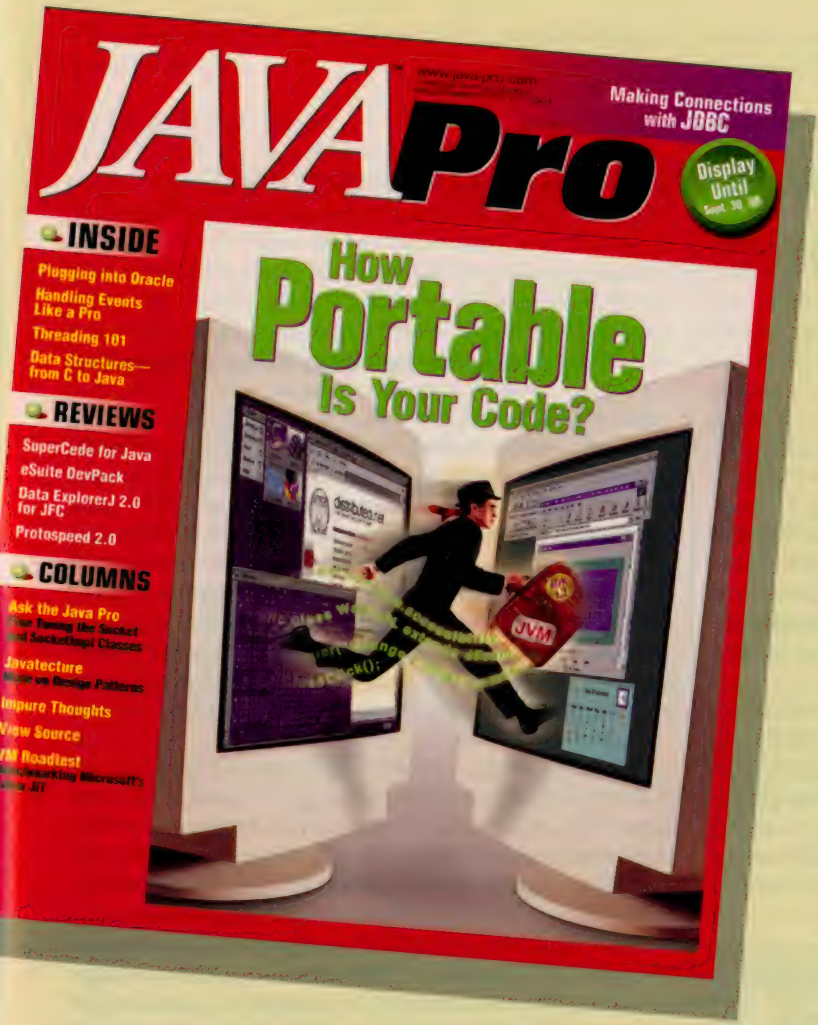
Creating Java and Win32 Memory-Mapped Files
Registered Locator+: **JP0399**
Premier Locator+: **JP0399TG**

Flag	Purpose
FILE_OPEN_EXISTING	Requires that file already exist or the open fails.
FILE_CREATE_NEW	If the file already exists, the creation fails.
FILE_OPEN_ALWAYS	If the file exists it is opened, otherwise a new file is created.
FILE_MAP_READ	Opens the file with read access and attempts to write to the file fail.
FILE_MAP_READWRITE	Opens the file for read and write access.
FILE_MAP_COPYONWRITE	Opens the file for read and write access. Any data written to the file is private to the process and will not effect the disk image or the data seen by other processes.

Table 1 Controlling the Base Class The MemoryMappedFile class uses this set of flags to provide precise control over the various permissions and other options the base class. They consist of a set of binary values that can be combined (via the | operator) into one set of flags.

Finally, a Magazine for Programmers Building Serious Java Business Applications

AND NOW JAVA™ Pro IS MONTHLY!



***Java Pro* magazine** is the first independent technical magazine dedicated to serious Java development.

Java Pro is the professional developer's independent source for information on building serious Java applications. *Java Pro* takes you beyond the hype to deliver hands-on, code-intensive, product-oriented information to help you harness Java's unique capabilities in your enterprise-wide development efforts. Each month, *Java Pro* tackles tough development issues and provides unbiased, no-nonsense coverage of the latest tools on the market.

Here's what you can expect from *Java Pro*:

- Practical techniques for enterprise development
- Tips and tricks for fine-tuning Java performance
- Independent product reviews and new-product announcements
- Complete source-code examples using JavaBeans, servlets, and more!

Don't miss this special offer: Save 58% off the cover price!

Call today to subscribe to *Java Pro*. You'll pay the low rate of $29.95 for a full year—twelve issues in all. That's a savings of more than 58% off the newsstand rate!

Java Pro comes to you from Fawcette Technical Publications, the creators of *Visual Basic Programmer's Journal* and *WEB Builder*. We provide technical information to more developers than anyone in the world.

Money-back guarantee

If you are not satisfied with ***Java Pro*** at any time, ***we will refund the cost of your subscription.*** No questions asked.

Call Today:

1.800.848.5523

Outside the U.S., call: **650.833.7100**

Order online: **http://www.java-pro.com**

FTP FAWCETTE TECHNICAL PUBLICATIONS
209 Hamilton Avenue
Palo Alto, CA 94301-2500

One year is twelve issues. Annual cover price is $71.40. Subscription begins upon receipt of payment. Please allow 4-6 weeks for delivery of first issue. International subscriptions must be payable in U.S. dollars plus postage. Canada/Mexico add $18/year; all other countries add $44/year for airmail. Canadian GST included.

Java is a trademark of Sun Microsystems, Inc.

Security for Servlets

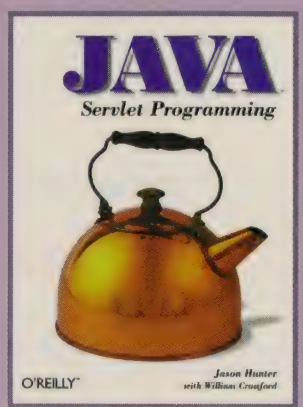
You can custom design the "sandbox" on your server with Java's many security options

by Jason Hunter with William Crawford

It's easy to imagine that our servlets exist in a perfect world where everyone is trustworthy and nobody locks their doors at night. Sadly, that's a fantasy world: The truth is that the Internet has its share of fiendish rogues. As companies place more emphasis on online commerce and load their intranets with sensitive information, security becomes one of the most important topics in Web programming.

Security is the science of keeping sensitive information in the hands of authorized users. On the Web, this boils down to three important issues: Authentication, being able to verify the identities of the parties involved; Confidentiality, ensuring that only the parties involved can understand the communication; and Integrity, being able to verify that the content of the communication is not changed during transmission.

This article is adapted from chapter 8 of *Java Servlet Programming*. Copyright 1998, by O'Reilly & Associates. Reproduced by permission. A'l rights reserved. *Java Servlet Programming*, covers the following topics: Retrieving Information, Sending HTML Information, Sending Multimedia Content, Session Tracking, Security, Database Connectivity, Applet-Servlet Communication, Interservlet Communication, Internationalization, Servlet API Quick Reference, and more. For ordering information, visiting www.oreilly.com.



A client wants to be sure that it is talking to a legitimate server (authentication), and it also want to be sure that any information it transmits, such as credit card numbers, is not subject to eavesdropping (confidentiality). The server is also concerned with authentication and confidentiality—if a company is selling a service or providing sensitive information to its own employees, it has a vested interest in making sure that nobody but an authorized user can access it. Both sides need integrity to ensure that whatever information they send gets to the other party unaltered.

Authentication, confidentiality, and integrity are all linked by digital certificate technology. Digital certificates allow Web servers and clients to use advanced cryptographic techniques to handle identification and encryption in a secure manner. Thanks to Java's built-in support for digital certificates, servlets are an excellent platform for deploying secure Web applications that use digital certificate technology. [For more information on digital certificates you can read "An Authentic Substitute for Passwords" by Boris Feldman, *Web Builder*, December 1998. See the Online Resources box at the end of this article for more information.]

Security is also about making sure that crackers don't gain access to the sensitive data on your Web server. Because Java was designed as a secure, network-oriented language, it is possible to leverage the built-in security features and make sure that server add-ons from third parties are almost as safe as the ones you write yourself.

This article introduces the basics of Web security in the context of using servlets. It also discusses how to maintain the security of your Web server when running servlets from untrusted third parties. This article takes a high-level approach because many of the topics require Web server-spe-

Listing 1 Snooping the Authorization Information In this code a simple servlet tells the client its name and what kind of authentication has been performed. To see this servlet in action, install it on your Web server and protect it with a basic or digest security scheme.

```
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class AuthorizationSnoop extends HttpServlet

public void doGet(HttpServletRequest req,
  HttpServletResponse res)
                    throws ServletException, IOException
  res.setContentType("text/html");
  PrintWriter out = res.getWriter();

  out.println("<HTML><HEAD><TITLE>Authorization
    Snoop</TITLE></HEAD><BODY>");

  out.println("<H1>This is a password protected
    resource</H1>");
  out.println("<PRE>");
  out.println("User Name: " + req.getRemoteUser());
  out.println("Authorization Type: " + req.getAuthType());
  out.println("</PRE>");
  out.println("</BODY></HTML>");
```

cific administration to implement—the servlets just tag along for the ride.

HTTP Authentication

The HTTP protocol provides built-in authentication support—called basic authentication—based on a simple challenge/response, username/password model. With this technique, the Web server maintains a database of usernames and passwords and identifies certain resources (files, directories, servlets, and so on) as protected. When a user requests access to a protected resource, the server responds with a request for the client's username and password. At this point, the browser usually pops up a dialog box where the user enters the information and that input is sent back to the server as part of a second authorized request. If the submitted username and password match the information in the server's database, access is granted. The whole authentication process is handled by the server.

Basic authentication is very weak. It provides no confidentiality, no integrity, and only minimal authentication. This is because passwords are transmitted over the network, thinly disguised by a well-known and easily reversed Base64 encoding. Anyone monitoring the TCP/IP data stream has full and immediate access to all the information being exchanged, including the username and password. Also, passwords are often stored on the server in clear text, making them vulnerable to anyone cracking into the server's file system. Although it's certainly better than nothing, sites that rely exclusively on basic authentication cannot really be considered secure.

Digest authentication is a variation on basic authentication. Instead of transmitting a password over the network directly, a digest of the password is used instead. The digest is produced by taking a hash (using the very secure MD5 encryption algorithm) of the username, password, URI, HTTP request method, and a randomly generated "nonce" value provided by the server. Both sides of the transaction know the password and use it to compute digests. If the digests match, access is granted. Transactions are somewhat more secure than they would be otherwise, because digests are valid for only a single URI request and nonce value. The server, however, must still maintain a database of the original passwords. And, as of this writing, digest authentication is not supported by most browsers.

HTTP authentication can be useful in low-security environments. For example, a site that charges for access to content—such as an online newspaper—is more concerned with ease of use and administration than lock-tight security, so HTTP authentication is often sufficient.

Retrieving Authentication Information

A server can retrieve information about the server's authentication using two methods: `getRemoteUser()` and `getAuthType()`. Listing 1 shows a simple servlet that tells the client its name and what kind of authentication has been performed (basic, digest, or another alternative). To see this servlet in action, install it on your Web server and protect it with a basic or digest security scheme. Because Web-server implementations vary, you'll need to check your server documentation for the specifics on how to set this up.

The problem isn't protecting the server from malicious CGI programmers, it's protecting it from careless CGI programmers

Custom Authorization

Normally, client authentication is handled by the Web server. The server administrator tells the server which resources are to be restricted to which users, and information about those users (such as their passwords) is made available to the server. This is often good enough, but sometimes the desired security policy cannot be implemented by the server. Maybe the user list needs to be stored in a format that is not readable by the server. Or maybe you want any username to be allowed, as long as it is given with the appropriate "skeleton key" password. In these situations, a servlet can be implemented so that it learns about users from a specially formatted file or a relational database; it can also be written to enforce any security policy. Such a servlet can even add, remove, or manipulate user entries—something that isn't supported directly in the Servlet API, except through proprietary server extensions.

A servlet uses status codes and HTTP headers to manage its own security policy. The servlet receives encoded authorization credentials in the `Authorization` header. If it chooses to deny those credentials, it sends the `SC_UNAUTHORIZED` status code and a `WWW-Authenticate` header that describes the desired credentials. A Web server normally handles these details without involving its servlets, but for a servlet to do its own authorization, it must handle these details, while the server is told not to restrict access to the servlet.

The `Authorization` header, if sent by the client, contains the client's username and password. With basic authoriza-

Listing 2 Security in a Servlet This servlet performs custom authorization and restricts access to its "top-secret stuff" to the users (and passwords) it recognizes in its user list. For this example, the list is kept in a simple Hashtable and its contents are hard-coded; this would be replaced with another mechanism, such as an external relational database, for a production servlet.

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class CustomAuth extends HttpServlet

Hashtable users = new Hashtable();

public void init(ServletConfig config) throws
  ServletException
  super.init(config);
  users.put("Wallace:cheese",     "allowed");
  users.put("Gromit:sheepnapper", "allowed");
  users.put("Penguin:evil",       "allowed");


public void doGet(HttpServletRequest req,
  HttpServletResponse res)
                  throws ServletException, IOException
  res.setContentType("text/plain");
  PrintWriter out = res.getWriter();

  // Get Authorization header
  String auth = req.getHeader("Authorization");

  // Do we allow that user?
  if (!allowUser(auth))
    // Not allowed, so report he's unauthorized
    res.sendError(res.SC_UNAUTHORIZED);
    res.setHeader("WWW-Authenticate", "BASIC realm=users");
    // Could offer to add him to the allowed user list

  else
    // Allowed, so show him the secret stuff
    out.println("Top-secret stuff");


// This method checks the user information sent in the
//Authorization header against the database of users
//maintained in the users Hashtable.
protected boolean allowUser(String auth) throws IOException
  if (auth == null) return false;  // no auth

  if (!auth.toUpperCase().startsWith("BASIC "))
  return false;  // we only do BASIC

  // Get encoded user and password, comes after "BASIC "
  String userpassEncoded = auth.substring(6);

  // Decode it, using any base 64 decoder
  sun.misc.BASE64Decoder dec = new sun.misc.BASE64Decoder();
  String userpassDecoded = new
    String(dec.decodeBuffer(userpassEncoded));

  // Check our user list to see if that user and password
  // are "allowed"
  if ("allowed".equals(users.get(userpassDecoded)))
    return true;
  else
    return false;
```

tion, the `Authorization` header contains the string of "`username:password`" encoded in Base64. For example, the username of "`webmaster`" with the password "`try2gueSS`" is sent in an `Authorization` header with the value:

```
BASIC d2VibWFzdGVyOnRyeTJndWVTUw
```

A servlet can send an `WWW-Authenticate` header to tell the client the authorization scheme and the realm against which users will be verified. A realm is a collection of user accounts and protected resources. For example, to tell the client to use basic authorization for the realm `Admin`, the `WWW-Authenticate` header is:

```
BASIC realm="Admin"
```

Listing 2 shows a servlet that performs custom authorization, receiving an Authorization header and sending the `SC_UNAUTHORIZED` status code and `WWW-Authenticate` header when necessary. The servlet restricts access to its "top-secret stuff" to those users (and passwords) it recognizes in its user list. For this example, the list is kept in a simple Hashtable and its contents are hard-coded; this would be replaced with another mechanism, such as an external relational database, for a production servlet.

To retrieve the Base64-encoded username and password, the servlet needs to use a Base64 decoder. Fortunately, there are several freely available decoders. For this servlet, we have chosen to use the `sun.misc.BASE64Decoder` class that accompanies the JDK. Being in the `sun.*` hierarchy means it's unsupported and subject to change, but it also means it's probably already on your system. You can find the details of Base64 encoding in RFC 1521 at http://ds.internic.net/rfc/rfc1521.txt.

Although the Web server is told to grant any client access to this servlet, the servlet sends its

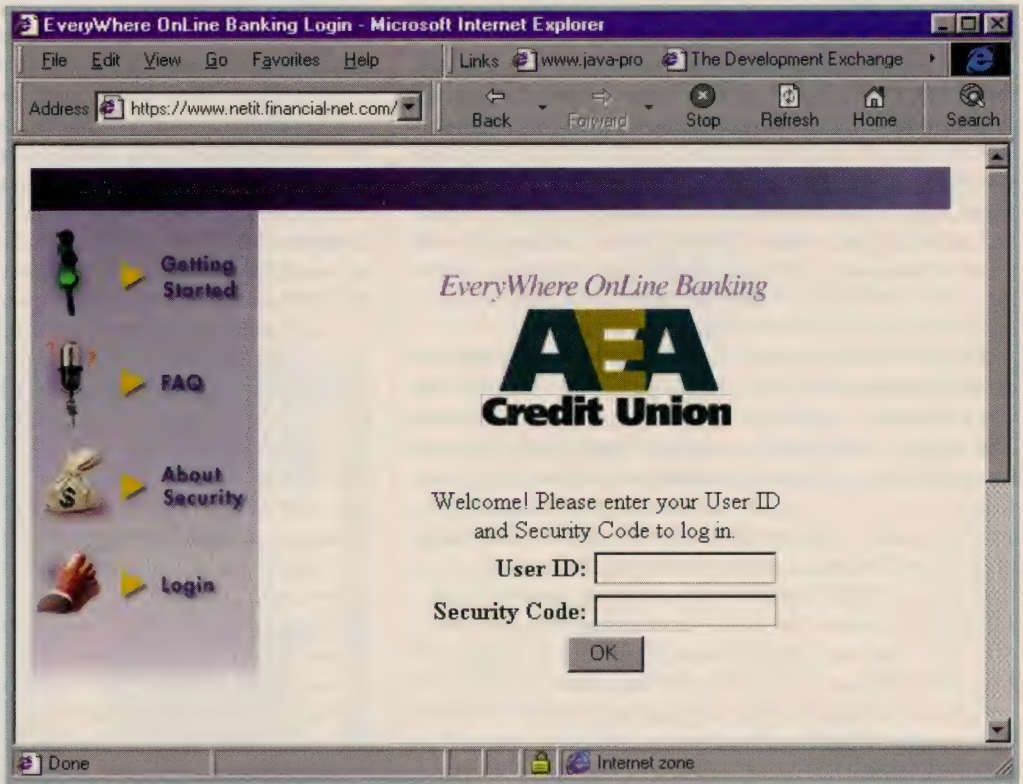


Figure 1 An Online Banking Login Screen This custom login form politely asks for specific banking credentials and submits the information to the code that validates the login. If the information passes security they are given access protected resources.

RSA Kills Hackers Dead!

So You Can Secure Your Applications With Confidence.

Don't let the threat of a few pesty hackers ruin your day. Build your applications with RSA Data Security components and you'll build in security that stops hackers dead in their tracks.

That's why RSA, a Security Dynamics Company, is the crypto-security source for more than 400 leading OEM companies. And why you'll find it in more than 300 million active installations worldwide, ranging from smart cards and cell phones to Web browsers and e-commerce systems.

RSA makes security a real picnic. Since 1982, RSA security technology has been the foundation of the information industry. Today, we offer a complete set of security components that allow you to create secure C and Java programs as well as SSL and S/MIME enabled applications.

As a developer working with RSA, you simply select the library components you need and build them into your application. And you can be sure RSA security will keep swarming hackers at bay.

Send the world (and its hackers) a message. The RSA name is proof to you and to users that your application will deliver information cleanly and securely. To learn more about RSA's complete package of data security solutions, please visit our website at **www.rsa.com/developersolutions**. And start developing some killer apps of your own.

Phone **650.295.7600**

RSA Data Security

RSA SECURE

This symbol lets you & your customers know that your application uses the best security technology.

RSA Data Security, RSA Secure, The RSA Secure logo, and the interlocking keys are trademarks of RSA Data Security, Inc. © 1999 RSA Data Security.

JAVA Pro™

Readers Poll

YOUR VIEWS ON *JAVA PRO*:

Your feedback is vital to us. Taking a minute to fill out this questionnaire will help us to refine our direction and content. This survey is also available online at www.java-pro.com. Tell us how you use Java and what you want from *Java Pro*. Thank you.

	Excellent	Good	Fair	Poor
Features				
Java in the Database	❑	❑	❑	❑
Using Memory-Mapped Files in Java	❑	❑	❑	❑
Security for Servlets	❑	❑	❑	❑
An Introduction to XML for Java Programmers	❑	❑	❑	❑
Columns				
Ask the Java Pro	❑	❑	❑	❑
Javatecture	❑	❑	❑	❑
Impure Thoughts	❑	❑	❑	❑
Servlet Solutions	❑	❑	❑	❑
VM Roadtest	❑	❑	❑	❑

TELL US ABOUT YOURSELF

1. **Do you subscribe to *Java Pro* magazine?** ❑ Yes ❑ No
2. **Are you a member of the Development Exchange's Premier Club?** ❑ Yes ❑ No

HOW DO YOU USE JAVA?

3. **Do you develop JavaBeans for use by others?**
   ❑ Yes, for internal use ❑ Yes, for commercial use ❑ No
4. **How would you describe your Java usage?**
   ❑ Just evaluating Java ❑ Small-scale development ❑ Large-scale development ❑ Not using
5. **What types of Java applications do you develop?**
   ❑ Browser-based Internet ❑ Browser-based intranet ❑ Stand-alone executable
   ❑ Server applications ❑ *n*-tier Client/Server
6. **What areas are you most interested in?**
   ❑ Database access ❑ CORBA ❑ XML/HTML ❑ Servlets ❑ Stand-alone apps
7. **Is developing "100" Pure Java applications important to you and/or your organization?**
   ❑ 100% Important ❑ 75% Important ❑ 50% Important ❑ Gimme J/Direct

Thank you for your help. Please photocopy this page and mail it or fax it to: **650-853-0230**

FAWCETTE TECHNICAL PUBLICATIONS • 209 HAMILTON AVE • PALO ALTO, CALIFORNIA 94301-2500 • USA

Listing ③ The login.html File Implementing a form-based custom authorization system is relatively straightforward with servlets. First, we need the login page. It can be written like any other HTML form. This code generates the form shown in Figure 2.

```
<HTML>
<TITLE>Login</TITLE>
<BODY>
<FORM ACTION=/servlet/LoginHandler METHOD=POST>
<CENTER>
<TABLE BORDER=0>
<TR><TD COLSPAN=2>
<P ALIGN=center>
Welcome!  Please enter your Name<br>
and Password to log in.
</TD></TR>

<TR><TD>
<P ALIGN=right><B>Name:</B>
</TD>
<TD>
<P><INPUT TYPE=text NAME="name" VALUE="" SIZE=15>
</TD></TR>

<TR><TD>
<P ALIGN=right><B>Password:</B>
</TD>
<TD>
<P><INPUT TYPE=password NAME="passwd" VALUE="" SIZE=15>
</TD></TR>

<TR><TD COLSPAN=2>
<CENTER>
<INPUT TYPE=submit VALUE="  OK   ">
</CENTER>
</TD></TR>
</TABLE>
```

top-secret output only to those users it recognizes. With a few modifications, it could allow any user with a trusted skeleton password. Or, like anonymous FTP, it could allow the "anonymous" username with any e-mail address given as the password.

Custom authorization can be used for more than restricting access to a single servlet. If we added this logic to our `ViewFile` servlet we could implement a custom access policy for an entire set of files. If we created a special subclass of HttpServlet and add this logic to that, we could easily restrict access to every servlet derived from that subclass. Our point is that with custom authorization, the security-policy limitations of the server do not limit the possible security-policy implementations of its servlets.

Form-Based Custom Authorization

Servlets can also perform custom authorization without relying on HTTP authorization, by using HTML forms and session tracking. It takes a bit more effort to give users a well-designed, descriptive, and friendly login page. For example, imagine you're developing an online banking site. Would you rather let the browser present a generic prompt for username and password or provide your customers with a custom login form that politely asks for specific banking credentials, as shown in Figure 1.

Many online services have chosen to use form-based custom authorization and implementing such a system is relatively straightforward with servlets. First, we need the login page. It can be written like any other HTML form. Listing 3 shows a sample login.html file that generates the form shown in Figure 2.

This form asks the client for their name and password, and submits the information to the `LoginHandler` servlet that validates the login. We'll see the code for `LoginHandler` soon, but first we should ask ourselves, "When is the client going to see this login page?" It's clear they can browse to this login page directly, perhaps following a link on the site's front page. But what if they try to access a protected resource directly without first logging in? In that case, they should be redirected to this login page and, after a successful login, be redirected back to the original target. The process should work as seamlessly as having the browser pop open a window. Listing 4 shows a servlet that implements this redirection behavior. It outputs its secret data only if the client's session object indicates they have already logged in. If they haven't logged in, the servlet saves the request URL in the session for later use, and then redirects them to the login page for validation.

This servlet sees if the client has already logged in by checking the session for an object with the name "`logon.isDone`." If such an object exists, the servlet knows that the client has already logged in and therefore allows them to see the secret goods. If it doesn't exist, the client must not have logged in, so the servlet saves the request URL under the name "`login.target`", and redirects the client to the login page. Under form-based custom authorization, all protected resources (or the servlets that serve them) have to implement this behavior. Subclassing, or the use of a utility class, can simplify this task.

Now for the login handler. After the client enters their information on the login form, the data is posted to the `LoginHandler` servlet shown in Listing 5. This servlet checks the username and password for validity, and if the client fails the check, they are told that access is denied. If the client passes, that fact is recorded in the session object and they are immediately redirected to the original target.

The actual validity check in this servlet is quite simple: It assumes any username and password are valid. That keeps things simple so we can concentrate on how the servlet behaves when the login is successful. The servlet saves the user's name (any old object will do) in the client's session under the name "`logon.isDone`", as a marker that tells all protected resources this client is okay. It then redirects the client to the original target saved as "`login.target`", seamlessly sending them where they wanted to go in the first place. If that fails for some reason, the servlet redirects the user to the site's home page.

Secure Sockets Layer (SSL)

The Secure Sockets Layer (SSL) sits between the application-level protocol (in this case HTTP) and the low-level transport protocol (for the Internet, almost exclusively TCP/IP). It handles the details of security management using public key cryptography to encrypt all client/server communication. SSL was introduced by Netscape with Netscape Navigator 1. It has since become the de facto standard for secure online communications and forms

Figure ② A Friendly Login Form This simple login form was generated from the HTML in Listing 3.

the basis of the Transport Layer Security (TLS) protocol currently under development by the Internet Engineering Task Force. (For more information on TLS, see www.consensus.com/ietf-tls.)

SSL Version 2.0 is the version first to gain widespread acceptance, and it includes support for server certificates only. It provides authentication of the server, confidentiality, and integrity. Here's how it works:

- A user connects to a secure site using the HTTPS (HTTP plus SSL) protocol. (You can detect sites using the HTTPS protocol because their URLs begin with https instead of http.)
- The server signs its public key with its private key and sends it to the browser.
- The browser uses the server's public key to verify that the same person who signed the key actually owns it.
- The browser checks to see whether a trusted certificate authority signed the key. If one didn't, the browser asks the user if the key can be trusted and proceeds as directed.
- The client generates an asymmetric (DES) key for the session, which is encrypted with the server's public key and sent back to the server. This new key is used to encrypt all subsequent transactions. The asymmetric key is used because of the high computational cost of public key cryptosystems.

All this is transparent to servlets and servlet developers. You just need to obtain an appropriate server certificate, install it, and configure your server appropriately. Information transferred between servlets and clients is now encrypted. Voilá, security!

SSL Client Authentication

Our security toolbox now includes strong encryption and strong server authentication, but only weak client authentication. Of course, using SSL 2.0 puts us in better shape because SSL-equipped servers can use the basic authentication methods discussed at the beginning of this article without concern for eavesdropping. We still don't have proof of client identity—after all, anybody could have guessed or gotten a hold of a client username and password.

Listing 4 A Protected Resource This servlet that implements redirection behavior so that when a user attempts to access a protected resource directly without first logging in, they are redirected to this login page and, after a successful login, are redirected back to the original target.

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class ProtectedResource extends HttpServlet

public void doGet(HttpServletRequest req,
  HttpServletResponse res)
                    throws ServletException, IOException
  res.setContentType("text/plain");
  PrintWriter out = res.getWriter();

  // Get the session
  HttpSession session = req.getSession(true);

  // Does the session indicate this user already logged in?
  Object done = session.getValue("logon.isDone");
    // marker object
  if (done == null)
    // No logon.isDone means he hasn't logged in.
    // Save the request URL as the true target and redirect
    // to the login page.
    session.putValue("login.target",
                HttpUtils.getRequestURL(req).toString());
    res.sendRedirect(req.getScheme() + "://" +
                req.getServerName() + ":" +
                  req.getServerPort() +
                    "/login.html");
    return;


  // If we get here, the user has logged in and can see
  // the goods
  out.println("Unpublished O'Reilly book manuscripts await
    you!");
```

Listing 5 Handling a Login This servlet checks the username and password for validity, and if the client fails the check, they are told that access is denied. If the client passes, that fact is recorded in the session object, and they are immediately redirected to the original target.

```
import java.io.*;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;

public class LoginHandler extends HttpServlet

public void doPost(HttpServletRequest req,
  HttpServletResponse res)
                    throws ServletException, IOException
  res.setContentType("text/html");
  PrintWriter out = res.getWriter();

  // Get the user's name and password
  String name = req.getParameter("name");
  String passwd = req.getParameter("passwd");

  // Check the name and password for validity
  if (!allowUser(name, passwd))
  out.println("<HTML><HEAD><TITLE>Access
    Denied</TITLE></HEAD>");
    out.println("<BODY>Your login and password are
      invalid.<BR>");
    out.println("You may want to <A HREF=/login.html>
      try again</A>");
    out.println("</BODY></HTML>");
  else
    // Valid login. Make a note in the session object.
    HttpSession session = req.getSession(true);
    session.putValue("logon.isDone", name);
    // just a marker object

    // Try redirecting the client to the page he first tried
    // to access
    try
      String target = (String)
        session.getValue("login.target");
      if (target != null)
        res.sendRedirect(target);
      return;

    catch (Exception ignored)

    // Couldn't redirect to the target. Redirect to the
    // site's home page.
    res.sendRedirect(req.getScheme() + "://" +
                  req.getServerName() + ":" +
                    req.getServerPort());


protected boolean allowUser(String user, String passwd)
  return true;  // trust everyone
```

www.enterprise-zone.com

Need Help Tackling the Enterprise?

Find all the information you need on DevX's enterprise-zone—

the most comprehensive independent source of technical information for Windows and interactive developers.

Join today and get immediate access to powerful tools, valuable code and DevX's vast resource of development information to help you produce advanced division-level applications, including:

- **Technical articles and code** from *Visual Basic Programmer's Journal*, *WEB Builder*, and *Java™ Pro* magazines
- **More Code**—download the best shareware and product demos
- **Programming tips**—plus a library of thousands more
- **Products**—find the right tool from a database of over 5,000 tools
- **Reviews of 100's of products**—with new reviews posted weekly
- **Dedicated Development forums**
- **Conference Updates**—register online for VBITS and the NT Enterprise Developer's Conference
- **Buy Programming Books Online** Save up to 40% on selected titles, plus get editors' picks and reviews of top programming titles

Enterprise-Zone is a member of the family of Windows and interactive development sites located on The Development Exchange at http://www.devx.com

DEV·X The Development Exchange

Windows and Visual Basic are registered trademarks of Microsoft Corporation. The Development Exchange and DevX are trademarks of Fawcette Technical Publications.
Fawcette Technical Publications, 209 Hamilton Avenue, Palo Alto, CA 94301-2500. Phone: 650.833.7100; Fax: 650.853.0230; www.devx.com

SSL 3.0 fixes this problem by providing support for client certificates. These are just like the certificates that servers use, but they are registered to clients instead. As of this writing, VeriSign claims to have distributed more than 750,000 client certificates. SSL 3.0 with client authentication works the same way as SSL 2.0, except that after the client has authenticated the server, the server requests the client's certificate. The client then sends its signed certificate, and the server performs the same authentication process as the client did, comparing the client certificate to a library of existing certificates (or simply storing the certificate to identify the user on a return visit). As a security precaution, many browsers require the client user to enter a password before they will send the certificate.

Once a client has been authenticated, the server can allow access to protected resources such as servlets or files just as with HTTP authentication. The whole process occurs transparently, without inconveniencing the user. It also provides an extra level of authentication because the server knows the client with a John Smith certificate really is John Smith (and it can know which John Smith it is by reading his unique certificate). The disadvantages of client certificates are that users must obtain and install signed certificates, servers must maintain a database of all accepted public keys, and servers must support SSL 3.0 in the first place. As of this writing, most servers do, including the Java Web Server.

Retrieving SSL Authentication Information

As with basic and digest authentication, all of this communication is transparent to servlets.

Security is the science of keeping sensitive information in the hands of authorized users

However, it is sometimes possible for a servlet to retrieve the relevant SSL authentication information. The java.security package has basic support for manipulating digital certificates and signatures. To retrieve a client's digital information, however, a servlet has to rely on a server-specific implementation of the request's `getAttribute()` method. Listing 6 shows how to use `getAttribute()` to fetch the details of a client's certificates. Remember that this works only for the Java Web Server. Other servlet implementations, if they include this functionality, are likely to do it in a slightly different way.

The first certificate is the user's public key. The second is VeriSign's signature that vouches for the authenticity of the first signature. Of course, the information from these certificate chains isn't particularly useful to the application programmer. In some applications, it is safe to assume that a user is authorized if they got past the SSL authentication phase. For others, the certificates can be picked apart using the javax.security.cert.X509Certificate class. More commonly, a Web server allows you to assign a username to each certificate you tell it to accept. Servlets can then call `getRemoteUser()` to get a unique username. This solution works with almost all Web servers.

Running Servlets Securely

CGI programs and C-based plug-ins operate with relatively unfettered access to the server machine on which they execute (limited on Unix machines by the user account permissions of the Web server process). This isn't bad for an isolated programmer developing for a single Web server, but it's a security nightmare for Internet

Listing 6 Examining Client Certificates To retrieve a client's digital information, a servlet has to rely on a server-specific implementation of the request's getAttribute() method. This code shows how to use getAttribute() to fetch the details of a client's certificates. Remember that this works only for the Java Web Server.

```
import javax.security.cert.X509Certificate;

out.println("<PRE>");

// Display the cipher suite in use
String cipherSuite =
(String) req.getAttribute("javax.net.ssl.cipher_suite");
out.println("Cipher Suite: " + cipherSuite);

// Display the client's certificates, if there are any
if (cipherSuite != null)
X509Certificate certChain[] =
  (X509Certificate[])
    req.getAttribute("javax.net.ssl.peer_certificates");
if (certChain != null)
  for (int i = 0; i < certChain.length; i++)
    out.println ("Client Certificate [" + i + "] = "
                 + certChain[i].toString());



out.println("</PRE>");

Here's the output we first saw in Chapter 4:
   Cipher Suite:  SSL_RSA_EXPORT_WITH_RC4_40_MD5
Client Certificate [0] = [
X.509v3 certificate,
Subject is
OID.1.2.840.113549.1.9.1=#160F6A68756E746572407367692E636F6D
  ,
CN=Jason Hunter, OU=Digital ID Class 1 - Netscape,
OU="www.verisign.com/repository/CPS Incorp. by
  Ref.,LIAB.LTD(c)96",
OU=VeriSign Class 1 CA - Individual Subscriber, O="VeriSign,
  Inc.", L=Internet
Key:  algorithm = [RSA], exponent = 0x    010001, modulus =
  b35ed5e7 45fc5328 e3f5ce70 838cc25d 0a0efd41 df4d3e1b
    64f70617 528546c8
  fae46995 9922a093 7a54584d d466bee7 e7b5c259 c7827489
    6478e1a9 3a16d45f
Validity  until
Issuer is OU=VeriSign Class 1 CA - Individual Subscriber,
  O="VeriSign, Inc.",
L=Internet
Issuer signature used [MD5withRSA]
Serial number =     20556dc0 9e31dfa4 ada6e10d 77954704
]
Client Certificate [1] = [
X.509v3 certificate,
Subject is OU=VeriSign Class 1 CA - Individual Subscriber,
  O="VeriSign,
Inc.", L=Internet
Key:  algorithm = [RSA], exponent = 0x    010001, modulus =
  b614a6cf 4dd0050d d8ca23d0 6faab429 92638e2c f86f96d7
    2e9d764b 11b1368d
  57c9c3fd 1cc6bafe 1e08ba33 ca95eabe e35bcd06 a8b7791d
    442aed73 f2b15283
  68107064 91d73e6b f9f75d9d 14439b6e 97459881 47d12dcb
    ddbb72d7 4c3f71aa
  e240f254 39bc16ee cf7cecba db3f6c2a b316b186 129dae93
    34d5b8d5 d0f73ea9
Validity  until
Issuer is OU=Class 1 Public Primary Certification Authority,
  O="VeriSign,
Inc.", C=US
Issuer signature used [MD2withRSA]
Serial number =     521f351d f2707e00 2bbeca59 8704d539
]
```

www.cplus-zone.com

Need C++ Programming Information *Now*?

It's all here on DevX's Cplus-zone—the most comprehensive independent source of technical information for Windows and interactive developers.

Join today and get immediate access to powerful tools, valuable code and DevX's vast resource of C++ programming information, including:

- **Technical articles and code** from the pages of *Visual Basic Programmer's Journal* and *WEB Builder* magazines
- **More Downloadable Code**—the best shareware and product demos to save you weeks of surfing!
- **Daily C++ programming tips**—plus a library of thousands more
- **Products**—find the right tool from a database of over 5,000 tools
- **Reviews of 100's of products**—with new reviews posted weekly
- **Dedicated C++ Development forums**
- **Buy Programming Books Online** Save up to 40% on selected titles, plus get editors' picks and reviews of top C++ titles

Cplus-Zone is a member of the family of programming sites located on The Development Exchange at http://www.devx.com

DEV·X The Development Exchange

Windows, Visual Basic, and Microsoft are registered trademarks of Microsoft Corporation. The Development Exchange and DevX are trademarks of Fawcette Technical Publications.
Fawcette Technical Publications, 209 Hamilton Avenue, Palo Alto, CA 94301-2500. Phone: 650.833.7100; Fax: 650.853.0230; www.devx.com

service providers (ISPs), corporations, schools, and anyone else running shared Web servers.

For these sites, the problem isn't just protecting the server from *malicious* CGI programmers. The more troublesome problem is protecting it from *careless* CGI programmers. There are dozens of well-known CGI programming mistakes that could let a malicious client gain unauthorized access to the server machine. One innocuous-looking but poorly written Perl `eval` function is all it takes. For an extensive list of CGI security gotchas, see chapter 6 of The WWW Security FAQ at www.w3.org/Security/Faq/www-security-faq.html.

To better understand the situation, imagine you're an ISP and you want to give your customers the ability to generate dynamic content using CGI programs. What can you do to protect yourself? Historically, ISPs have chosen one of three options:

- Have blind faith in the customer. He's a good guy and a smart programmer, and besides, we have his credit-card number.
- Educate the customer. If he reads the WWW Security FAQ and passes a written test, we'll let him write CGI programs for our server.
- Review all code. Before we install any CGI program on the server, we'll have our expert review it and scan for security problems.

None of these approaches works very well. Having blind faith is just asking for trouble. Programmer education helps, but programmers are human and bound to make mistakes. As for code review, there's still no guarantees, plus it takes time and costs money to do the extra work.

Fortunately, with servlets there's a better solution. Because servlets are written in Java, they can be forced to follow the rules of a security manager (or access controller with JDK 1.2) to greatly limit the server's exposure to risk, all with a minimal amount of human effort.

The Servlet Sandbox

Servlets built using JDK 1.1 generally operate with a security model called the "servlet sandbox." Under this model, servlets are either trusted and given open access to the server machine, or they're untrusted and have their access limited by a restrictive security manager. The model is very similar to the "applet sandbox," where untrusted applet code has limited access to the client machine.

A security manager is a class subclassed from java.lang.SecurityManager that is loaded by the Java environment to monitor all security-related operations: opening network connections, reading and writing files, exiting the program, and so on. Whenever an application, applet, or servlet performs an action that could cause a potential security breach, the environment queries the security manager to check its

TechTIP

Multithreading Applets—Correctly

The canonical design of multithreaded applets is inherently flawed. You know what I'm talking about: the `run()` method with the `repaint()/sleep()` statements embedded in a `while(true)`. Any self-respecting operating-systems instructor would fail that design with reckless abandon. Here's the offensive code:

```
public void run() {
  while (true) {
    repaint();
    try {
     Thread.sleep(100);
    } catch (InterruptedException e) {}
  }
}
```

Of course, we also have a `paint()` method somewhere, which does all our impressive applet drawing. What's wrong with this model? The problem is in the `sleep()` statement. Simply speaking, there is no perfect number to put in the `sleep()` statement. It's there to hold off calling `repaint()` until we know the VM inside our browser (or appletviewer) completed a call to our `paint()` method. Unfortunately, you can never anticipate how long that will take. That is dependent on the machine, OS, length of the `paint()` method, and whatever else happens to be going on in the OS at the same time. Even if you found a number that seemed to work well, it may well fail if the OS gets busy with other transient functions, and it may be disastrous when moved to a less powerful machine.

The bottom line is that you are trying to synchronize two threads with that `sleep()` statement. The two threads are your "run" thread and the thread the browser sends in to call your `paint()`. That is synchronization on a "guess" (Dr. Tanenbaum, please forgive us). This is seriously wrong.

How can we solve this problem correctly? In other words, how can I guarantee that only one `repaint()` is called for exactly one `paint()` call? Try this:

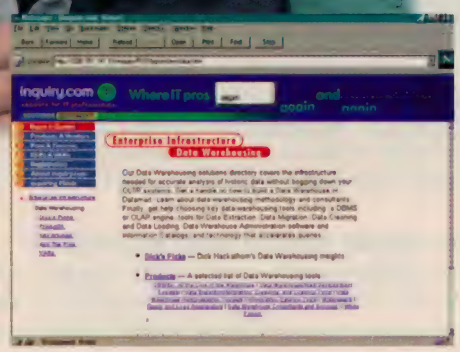
```
public synchronized void run() {
  while (true) {
    repaint();
    try {
     wait();
     Thread.sleep(100); // optional
    } catch (InterruptedException e) {}
  }
}

public void paint(Graphics g) {
    // drawing
    synchronized(this) {
      notifyAll();
    }
  }
```

What have we done? If you follow the flow, you'll see that we issue a `repaint()` and then wait. The `run()` thread cannot continue until the paint method is done drawing and as its last act, notifies waiting threads. This paradigm guarantees one `repaint()` per paint. We can still put in a `sleep()` to control how much CPU a possibly overzealous applet may take, but it is by no means synchronizing the threads.

—Paul Tyma

SEND YOUR TIP

If it's cool and we publish it, we'll pay you $25. Limit Code examples to about 20 lines. Be sure to include a clear explanation of what it does and why it's useful. Please e-mail to javaproedit@fawcette.com, or mail it to Java Tips, Fawcette Technical Publications, 209 Hamilton Avenue, Palo Alto, CA, 94301-2500, USA. Please include your mailing address.



"We found the IT answers we needed at inquiry.com."

When you need answers on development tools and techniques, turn to www.inquiry.com.

Our editors and "Ask the Pro" professionals provide unique insights on real-world development problems.

Now that inquiry.com is part of The Development Exchange suite of Web sites, you also have access to the largest database of independent product reviews, technical tips, and our Buyers Guide with information on over 5,000 tools for professional developers.

Nothing matches the efficiency and scope of www.inquiry.com.

Get the IT answers you need, free, today.

Find us at www.inquiry.com

inquiry.com
answers for IT professionals

© Inquiry.com, the inquiry.com logo, and The Development Exchange are trademarks of Fawcette Technical Publications.

Permission	Description
Load servlet	Allow the servlet to load a named servlet.
Write files	Allow the servlet to write any file on the local file system.
Listen to socket	Allow the servlet to accept incoming socket (network) connections.
Link libraries	Allow the loading of native libraries, such as the JDBC-ODBC bridge.
Read files	Allow the servlet to read any file on the local file system.
Open remote socket	Allow the servlet to connect to an external host.
Execute programs	Allow the servlet to execute external programs on the server. This is useful for servlets that absolutely require access to some system utilities, but it is very dangerous: rm and del qualify as an external programs!
Access system properties	Grant access to java.lang.System properties.

Table 1 Ask Nicely The Java Web Server includes eight permissions that can be granted to servlets.

Criterion	Factor
The servlet itself	This servlet can read files and load native libraries but cannot write files.
The client user	Any servlet responding to a request from this client user can write files.
The client host	Any servlet responding to a request from this machine can establish network connections.
Digital signatures	Any servlet in a JAR file signed by this entity has full reign on the server system.

Table 2 Permission Granted These are just some of the criteria that can be used to determine permission granting.

permissions. For a normal Java application, there is no security manager. When a Web browser loads an untrusted applet over the network, however, it loads a very restrictive security manager before allowing the applet to execute.

Servlets can use the same technology, if the Web server implements it. Local servlets can be trusted to run without a security manager, or with a fairly lenient one. For the Java Web Server 1.1, this is what happens when servlets are placed in the default servlet directory or another local source. Servlets loaded from a remote source, on the other hand, are by nature suspect and untrusted, so the Java Web Server forces them to run in a very restrictive environment where they can't access the local file system, establish network connections, and so on. All this logic is contained within the server and is invisible to the servlet, except that the servlet may see a SecurityException thrown when it tries to access a restricted resource. The servlet sandbox is a simple model, but it is already more potent than any other server extension technology to date.

Using digital signatures, it is possible for remotely loaded servlets to be trusted just like local servlets. Third-party servlets are often packaged using the Java Archive (JAR) file format. A JAR file collects a group of class files and other resources into a single archive for easy maintenance and fast download. Another nice feature of JAR files that is useful to servlets is that they can be signed using digital certificates. This means that anyone with the public key for "Crazy Al's Servlet Shack" can verify that their copy of Al's Guestbook Servlet actually came from Al.

Information transferred between servlets and clients is now encrypted. Voilá, security!

On some servers, including the Java Web Server, these authenticated servlets can then be trusted and given extended access to the system.

Fine-Grained Control

This all-or-nothing approach to servlet permissions is useful, but it can be overly limiting. Consequently, some servlet engines have begun to explore a more fine-grained protection of server resources, for example, allowing a specific servlet to establish a network connection but not write to the server's file system. This fine-grained control is fairly awkward using the JDK 1.1 notion of a `SecurityManager` class and therefore, isn't widely implemented, although the Java Web Server 1.1 proves it can be done. The Java Web Server 1.1 includes eight permissions that can be granted to servlets (see Table 1). Theoretically, any criterion can be used to determine what a servlet can or cannot do. It's possible for the security manager to base its permission-granting decision on any factor, including those shown in Table 2.

Access Controllers

JDK 1.2 introduces a new extension to the security manager system: the access controller. The new architecture is quite similar to the "give particular servlets particular privileges" approach implemented by the Java Web Server 1.1, except that it applies to all JDK 1.2 programs and therefore makes fine-grained permission implementations much easier.

An access controller allows what might be called super-fine-grained permission control. Instead of granting a servlet the general ability to write files, a servlet can be given the right to write to a single file—perfect for a counter servlet, for example. Or it can be given the right to read and write files only in the client user's home directory on the server—perfect for a client/server application. With access controllers, servlets can be given the rights to do exactly what they need to do and nothing more.

Access controllers work by placing individual pieces of code, often identified by digital signatures, into particular virtual domains. Classes in these domains can be granted fine-grained permissions, such as the ability to read from the server's document root, write to a temporary directory, and accept socket connections. All permission policy decisions are managed by a single instance of the `java.security.AccessController` class. This class bases its policy decisions on a simple configuration file, easily managed using a graphical user interface.

Instead of relying on complicated custom security managers as the Java Web Server team had to do, a servlet engine requires only a few lines of code to use an access controller. While the Java Web Server was the only servlet implementation supporting fine-grained security as of early 1998, once JDK 1.2 becomes popular, it should be easy for other servlet engine implementers to add the same level of access control. ■

Code Online

All the code in this article is available for free at www.java-pro.com. Use the Locator+ codes shown below to download the code. Free online registration is required. Members of the Premier Club can also read more about digital certificates in An Authentic Substitute for Passwords. Go to www.webbuildermag.com, click on the Web Builder archives and look at the December issue. See the Online Resources box in "Letters to the Editor" for more information.

Security for Servlets
Registered Locator+: **JP0399**
Premier Locator+: **JP0399JH**

www.java-zone.com

100% Pure Development Solutions. Delivered Daily Online.

It's all here on DevX's Java Zone.

Jump on today and get immediate access to the most comprehensive source of expert advice, information, code, and essential resources to help you develop serious Java applications, including:

- **Technical articles and code** from *Java™Pro, Visual Basic Programmer's Journal* and *WEB Builder* magazines
- **Daily programming tips**—plus a library of thousands more
- **Download Java tools**—new applications and applets featured weekly
- **Search Products**—from our database of over 5,000 programming tools
- **Product Reviews**—hundreds of reviews updated weekly
- **Forums**—dedicated to Java development
- **40% off Books of the Week**—plus editors' picks and reviews of top programming titles
- **Conference updates**—register online

And that's just the beginning.

DEV·X
The Development Exchange

Explore the rest yourself at
www.java-zone.com.

Java-Zone is a member of the family of Windows and interactive development sites located on The Development Exchange at http://www.devx.com.
Java is a registered trademark of Sun Microsystems. The Development Exchange and DevX are trademarks of Fawcette Technical Publications.
Fawcette Technical Publications, 209 Hamilton Avenue, Palo Alto, CA 94301-2500. Phone: 650.833.7100; Fax: 650.853.0230.

An Introduction to XML for Java Programmers

The Extensible Markup Language and Java are a perfect duo for data exchange and more

by Piroz Mohseni

In the computing world, every once in a while a new technology comes along that promises to make dramatic changes to our computing tasks. Extensible Markup Language (XML) has been heralded as one such technology. I have no intention of adding to the hype surrounding XML, but I believe there is something profound behind the three letters: X, M, and L.

XML has often been associated with HTML and discussed in the context of content markup, content management, search engines, and meta languages, but there are some aspects of XML that are also a perfect match for Java. And, not surprisingly, there are standards that link the two technologies. The goal of this article is to show what XML is and how it can be used in the context of Java. XML is not a programming language—it is merely a language in which you can create a markup language. XML provides a method to represent data in a universal way. Data and the logic to process data (programs) have a long-standing relationship. It should come as no surprise that such a relationship exists and will emerge fully from XML and Java.

What is XML?

XML is a markup language. In that regard, it might seem to be like HTML or any other language that ends in "ML." XML, however, has one fundamental difference—extensibility. With HTML, you are given a set of tags (markups) and you are bound by the given set as you markup a document. With XML, *you* invent the tags, so you can use XML to markup different types of documents for very specific purposes.

With a few exceptions, most tags in HTML are for formatting purposes. The tags instruct the browser how a certain piece of text ought to be displayed. The browser understands a finite number of tags. Any tags it does not recognize cannot be validated, so the browser ignores them. As a markup language, HTML is a subset of XML: All the tags used by HTML can be described using XML.

A document can also be marked up to describe its content. Have you ever used a search engine and had to sort through a maze of documents that had absolutely nothing with your specified search criteria? That's where marking up the content can make a big difference, and that's where XML comes in. When the engine encounters the word "apple," it should know whether the document is referring to apple as a fruit or as the computer company. With XML you can define the difference.

Let's take a look at a simple example of an XML document. The extensibility of XML should be evident:

```
<?xml version="1.0"?>

<name>
```

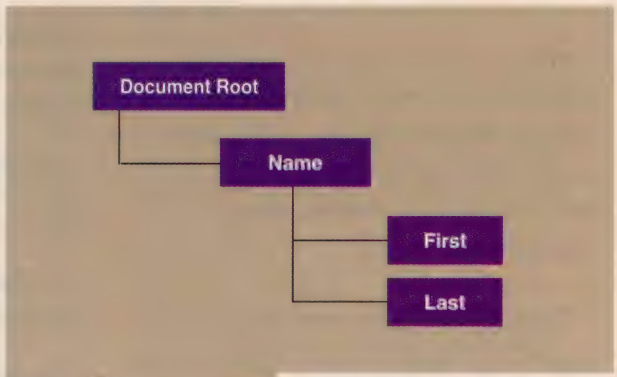


Figure 1 XML Hierachy The hierarchy of XML can be easily expressed in terms of a tree data structure. The root level is referred to as the document root. The first branch is the name entity, which itself is composed of two other entities

Piroz Mohseni is a Senior Managing Consultant with Automated Concepts Incorporated (ACI) in New York. He specializes in enterprise Java, XML, and e-commerce applications. Reach him at piroz.mohseni@autocon.com.

```
  <first>John</first>
  <last>Doe</last>
</name>
```

We created the tags <name>, <first>, <last> to describe an entity—a person's name. These tags were not part of any standard except one which we created. New standards are constantly emerging. For example, the Chemical Markup Language (CML) is used to describe chemical elements. Such standards are also referred to as vocabularies or grammars for specific domains. Table 1 lists some of the current XML standards.

Our simple example also displays XML's structure. The markup provides a structured way to express the name entity. The structure we created shows that a name entity is composed of two parts: a first name and a last name. It also shows the "nested" relationship between these entities. For example, the first and last names are nested inside a name entity. The ability of XML to add structure to a given document is very important and sets the foundation for interactions between a programming language such as Java and XML documents.

The hierarchy represented in our example can be easily expressed in terms of a tree data structure as shown in Figure 1. The root level is referred to as the document root. The first branch is the name entity, which itself is composed of two other entities. The Document Object Model (DOM), which we'll discuss later, is based on a similar tree structure.

Markup Language	Function
Synchronized Multimedia Integration Language (SMIL)	Multimedia documents such as movies, sounds, pictures.
Mathematical Markup Language (MathML)	Developed to describe mathematical equations.
Bioinformatic Sequence Markup Language (BSML)	Encoding and displaying DNA, RNA, and protein sequences.
Channel Definition Format (CDF)	Push technology implemented by Microsoft intended to push Web content to browsers.
Resource Description Framework (RDF)	Developed to improve interoperability of applications in terms of metadata property sets allowing search engines to search for terms in a specific context.
Open Software Description Format (OSD)	Standardized software distribution, version control and patch control.
Chemical Markup Language (CML)	Developed to describe chemical elements.
Open Financial Exchange	A broad-based framework for exchanging financial data and instructions between customers and their financial institutions. Sponsored by Microsoft, Intuit, and CheckFree.
Web Interface Definition Language (WIDL)	Submitted by WebMethods, this specification is intended to facilitate interaction with existing HTML documents and future XML documents via the Web.

Table 1 XML Standards The extensibility of XML makes it possible to customize a language for specific use. This table defines some of the current standards and their use.

Self Validation

A unique feature of XML documents is that that they are self validating—the rules of the standard used to create the document can themselves be part of the document. Imagine a set of data with a piece describing whether the data is valid or not. In the XML world, there are two types of validity. The first type deals with the physical structure of the document. The example we showed earlier passes the validity test at this level. A document that meets the criteria is called "well-formed." Such documents start with the line specifying the XML version and are then composed of a series of tags. For each opening tag, there must be a closing tag, and when you nest the tags, they have to be hierarchical. As long as these basic structural issues are taken care of, the document is well-formed. (That's not the formal definition for well-formed, but it is adequate for our discussion.)

In the second level of validation, a Document Type Definition (DTD) accompanies the document. The DTD, can be part of the document or an external document. The DTD specifies the rules or the grammar of what the XML document contains. For example, with the DTD you can specify that the <name> tag must have two nested tags, namely <first> and <last>. XML tags can also have attributes like most HTML tags. For instance, the <form> tag in HTML usually contains two attributes, method and action: <form method="post" action="foo.cgi">

With DTD, you can specify what attributes a given tag must have and which attributes are optional. You also have some control over the data type of the attribute. Although you can get away with not including the attribute value in quotation marks in HTML, in XML you can't. Our simple XML example is revisited here with the DTD as part of the document. (If the DTD is external to the document, a link will be specified.)

```
<?xml version="1.0"?>
<!DOCTYPE name [
    <!ELEMENT name(first, last)>
]>
<name>
   <first>John</first>
   <last>Doe</last>
</name>
```

This grammar specifies that the name entity must include two other entities called "first" and "last." If someone using our grammar fails to include one of the tags or mistypes their name, then the XML parser can catch that error. Documents that conform to the grammar specified by their respective DTD are known as "valid" documents.

There are some aspects of XML that are a perfect match for Java

In order to become familiar with the structure of XML documents, I suggest you experiment with an XML editor. Try XML Notepad, currently in beta, from Microsoft or XML Pro 1.2 from Vervet Logic (see "Resources"). The demo version of XML Pro includes a nice tutorial on XML and DTDs.

The Java Mix

The ability to define data structures in plain text make XML useful as a data exchange mechanism among applications.

But what does this have to do with Java? Just as a browser brings HTML to life, Java can bring XML documents to life. An XML document by itself is merely a collection of data and metadata. A programming language is required to process that data and perform useful operations using the data. In the remainder of this article, we'll take a look at how Java can leverage XML documents. It should be noted that XML is inherently language independent. Other languages such as Perl, Python, and C++ can also be used to manipulate XML documents.

An XML document is a text file. Initially you might be tempted to use Java to read and write to the file as it would to any other text file. Although this type of interaction is possible, it doesn't take advantage of the benefits of XML. As we have seen, valid XML documents have a very specific tree structure. It makes sense to preserve that structure and take advantage of it in any interaction with the XML document. Programs that interact with XML documents are known as XML parsers. There are a number of implementations out there from Microsoft, IBM, and Sun (see "Resources"). In this article, we'll use the one from Sun. I suggest that you download and install the XML Library from Sun, available at http://developer.java.sun.com/developer/earlyAccess/xml before proceeding. This library includes examples, documentation, and Sun's implementation of DOM and the SAX API. Sun refers to the event-based API as the "parsing model" and the tree-based API as the "object model." You also need a JDK 1.1 compliant compiler (including JDK 1.2). A basic XML parser can ensure that check a document is well-formed. A more useful parser would also process the DTD and verify the validity of the XML document.

Now that we have established a framework for interaction between a program and an XML document, we can discuss some specifics. Given the tree-based structure of an XML document, one approach would be a tree-based API that maps an XML document into specific objects in a tree-like structure and allows for writing additional nodes and manipulation of existing nodes. This API must also allow for manipulation of the DTD that controls the data tree. The standard tree-based API is the Document Object Model (DOM). Sun's implementation is in the `com.sun.xml.tree` package.

Because every tree is navigable, you can also use an event-based API. In this environment, the data tree representing the XML document is navigated from its root level down. As each tag is encountered, an event is generated. The API allows a program to write "event handlers" to respond to these various events. This is similar to how GUI programs work. The standard event-based API is Simple API for XML (SAX). SAX is attractive for handling large documents because it is not required to load the entire document tree in memory to process it. Sun's implementation is in the `com.sun.xml.parser` package.

I will present two simple examples in this article. (More complete examples are available in the XML Library.) In the first example, I'll show how to read an XML document inside a Java program, parse it, and then write it out. In the second example, I'll create an XML document from scratch and then write it out into an XML file.

The program in Listing 1 takes an XML filename as input and sets that as the "input source" to the parser. It will be helpful to have the Java API documentation available while reviewing this code. The `createXmlDocument()` method actually parses the file. By giving a true or false parameter, you can instruct the parser to check for valid XML or a well-formed document. Once you have the file in memory, you can experiment with various methods available in the API, which are described in the documentation (included with the download). For our purposes, we'll write out what we have in memory using the `write()` method. Although this program doesn't do much, it demonstrates some very basic concepts about manipulating an XML document with Java. It also provides good skeleton code with which experiment. You can use the sample XML file or create your own.

Here is the output you would get with the program in Listing 1 if you use the example XML from the first part of the article:

```
D:\xml\xml-ea2\examples\article>java
  simpleXml name.xml
```

Resources

XML Editors

Microsoft XML Notepad; Microsoft: www.microsoft.com/xml/notepad/intro.asp

XML Pro; Vervet Logic: www.vervet.com

XML Parsers

Microsoft XML Parser in Java; Microsoft: www.microsoft.com/xml/parser/jparser.asp

XML Parser for Java; IBM: www.alphaworks.ibm.com

Sun Parser (Project X); Sun Microsystems: http://java.sun.com/products/javaprojectx/

More XML Info:

http://developer.java.sun.com/developer/earlyAccess/xml

www.xmlrepository.com

www.xmlinfo.com

www.ucc.ie/xml

www.oasis-open.org/cover/xml.html

www.w3.org/TR/1998/REC-xml-19980210.html

Listing 1 Manipulating XML in Java This program takes an XML filename as input and sets that as the "input source" to the parser, demonstrating some basic concepts of manipulating an XML document with Java.

```
import java.io.File;
import com.sun.xml.parser.*;
import com.sun.xml.tree.*;
import org.xml.sax.*;

public class simpleXml
{
    public static void main (String argv [])
    {
  InputSource in;
  XmlDocument xmlDoc;

// check to make sure an XML file is provided as input
  if (argv.length != 1) {
      System.err.println ("Usage: java simpleXml filename");
      System.exit (1);
  }

  try {
//static methods to create XML input sources from the
//given file
      in = Resolver.getSource (new File (argv [0]));

// parse it
      xmlDoc = XmlDocumentBuilder.createXmlDocument
        (in, false);

// print it out
      xmlDoc.write (System.out);

  } catch (SAXParseException e) {
      System.out.println ("Parsing error"
    + ", Line " + e.getLineNumber ()
    + ", Uri " + e.getSystemId () );

  } catch (SAXException e) {
      System.out.println(e);
  }
    catch (java.io.IOException e) {
      System.out.println(e);
  }
  System.exit (0);
    }
}
```

Coming Fall 1999—A Conference for the Technology Elite

Enterprise DEVELOPMENT 2000 CONFERENCE

Enterprise Computing's Peek Experience

Join acclaimed digital newsletter editor and publisher

David Coursey

as he guides you through the maze of distributed computing technologies to only the best new stuff. Middleware, servers, modeling, applications—it's all here. See what's in store for the year 2000 when you get a look behind the curtain with FTP and David Coursey.

Coming next fall from an alliance of Fawcette Technical Publications and UPSIDE: **Enterprise Development 2000**, a conference for the technology elite. David Coursey, editor of the widely-quoted newsletter *coursey.com*, was recognized by Adweek's *Marketing Computers* magazine as the #1 newsletter editor in the computer industry three years running.* His insights and sometimes biting commentary on the computer industry have made him one of the most often quoted observers in the industry.

Call today for more information

1-800-848-5523

or visit us online at
www.enterprisedev.com

PEOPLE • TECHNOLOGY • CAPITAL
UPSIDE
WWW.UPSIDE.COM

[c]
coursey.com

FTP FAWCETTE TECHNICAL PUBLICATIONS

209 Hamilton Avenue • Palo Alto, CA 94301-2500
Phone (650) 833-7100 • Fax (650) 853-0230
www.fawcette.com • www.devx.com

*from *Marketing Computers* June 1998, 1997 and May 1996 annual rankings of "Most Influential Computer Journalists"

Enterprise Development is a trademark of Fawcette Technical Publications.
UPSIDE magazine is a trademark of Upside Media Inc. and coursey.com is a trademark of David Coursey.

Listing ② Starting from Scratch This code creates a simple XML document and then writes it out. It uses methods that help you navigate a given tree.

```
import java.io.*;
import com.sun.xml.tree.*;
import org.w3c.dom.*;

public class domXml
{
    public static void main (String argv [])
    throws IOException, DOMException
    {
  XmlDocument xmlDoc = new XmlDocument ();
  ElementNode name = (ElementNode) xmlDoc.createElement
    ("name");
  ElementNode first = (ElementNode) xmlDoc.createElement
    ("first");
  ElementNode last = (ElementNode) xmlDoc.createElement
    ("last");

  Writer  out = new OutputStreamWriter (System.out);

    out.write ("No tree exists yet\n");
    out.write ("\n");

        xmlDoc.appendChild(name);
    name.appendChild (first);
    first.appendChild (xmlDoc.createTextNode("\n John \n"));
    name.appendChild (last);
        last.appendChild (xmlDoc.createTextNode("\n Doe
\n"));

    name.setAttribute ("city", "New York");

    out.write ("DOCUMENT is:\n");
    xmlDoc.write (out);
    out.write ("\n");
    out.flush ();

    System.exit (0);
      }
}
```

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE name [
]>

<name>
  <first>John</first>
  <last>Doe</last>
</name>
```

In our second example (see Listing 2), we'll create a simple XML document from scratch and then it write out. This exercise will familiarize you with some of the navigational methods that you will be using to go through a given tree. Note that we are not using the parser in this case. Following the code, we first create an instance of XmlDocument. We'll also create several ElementNode objects representing the name, first, and last tags. Initially, there is no tree, and we print a message to that effect. Then, using the `appendChild()` method, we create the hierarchy that represents our name structure. Although not part of the original XML document, we also set an attribute `city="New York"`. There are methods for writing out the DTD as well, but we won't use them here. You can use this code as a skeleton to practice creating more complex XML documents from scratch. Here is the output:

```
D:\xml\xml-ea2\examples\article>java
domXml
No tree exists yet

DOCUMENT is:
<?xml version="1.0" encoding="Cp1252"?>

<name city="New York">
  <first>
 John
</first>
  <last>
 Doe
</last>
</name>
```

More on XML

There is obviously more to XML and Java than what we have covered in this article. There are a number of XML technologies in development today that could dramatically change the way developers work. The SAX model will almost exclusively be used to parse large documents. Sometimes visualization of an XML tree becomes difficult. The Jtree in the Java Swing package can be used to display the structure of an XML document as it is being created or parsed (there are examples in Sun's XML Library). A more general solution would be a JavaBean that can be plugged into other applications needing a visual way to display an XML document.

Another area of interest is transportation of XML documents. The HTTP protocol is used to transfer HTML documents and seems like a good match for transporting XML documents. XML over HTTP is simple and robust and also allows for a smooth transition from the HTML-based sites to an XML-based site. From an application point of view, XML over HTTP offers a similar functionality to what CORBA or RMI provide in terms of message exchange among objects. Although the solution might not be as general-purpose as that of CORBA, in certain situations, the simplicity wins over the extra functionality.

There are bound to be more helper packages to simplify manipulation of XML documents within a particular context. Current implementations of DOM and SAX offer additional functionality not supported by all parsers. In the near future I expect the standards to become more focused and the implementations to converge toward a single standard. Standards such as Xpointers (used to navigate an XML tree and access different nodes and elements) and XML Linking Language (XLink) will also become available as part of the Java API.

Although the XML document represents a portable manner to encapsulate data, a programming language such as Java is needed to interact with that data. The basic premise of Java interaction with XML documents is divided into two sets of APIs. The first, DOM, is based on the tree-like structure of XML documents. The second is based on an event model where the parser goes through the document and generates events as it sees various tags. The program will then take care of the events using event handlers.

XML is an exciting technology and it has great potential, but it is still in the early stages. The standards surrounding XML must mature and the various implementations must converge towards the standards before we can receive the full benefit of this technology. Given the universal nature of XML and the portability associated with Java, they seem to be a good match. Java programs have been referred to as "plug-and-play." Perhaps XML has the potential to create "plug-and-play" data. ■

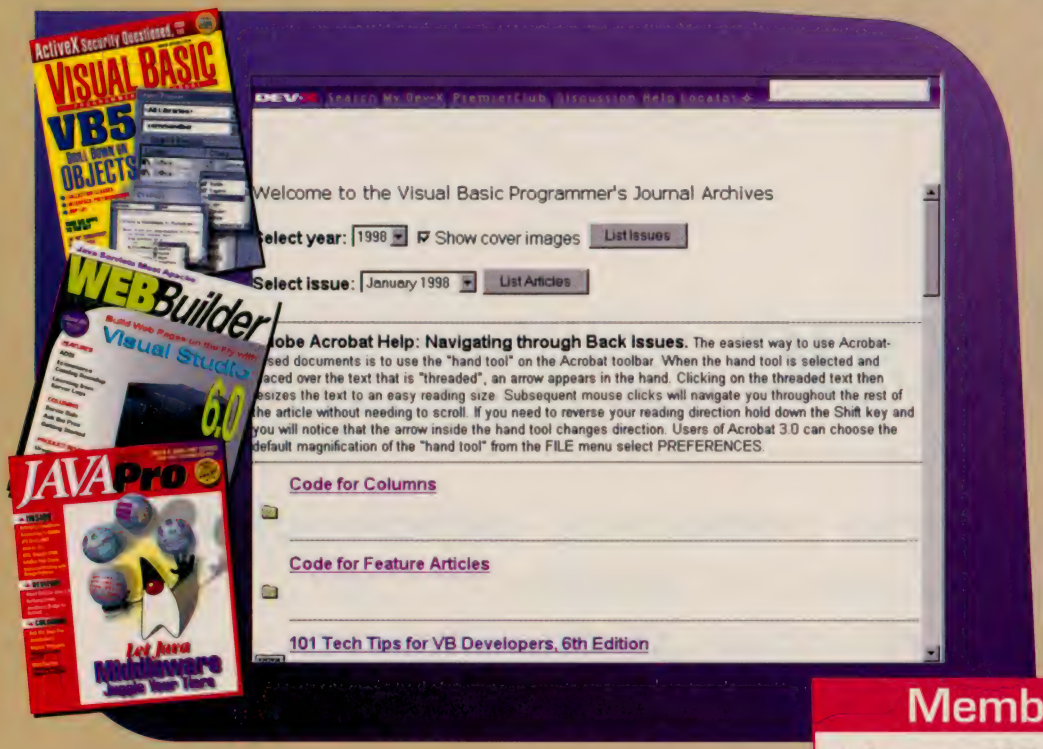
Code Online

All the code in this article is available for free at www.java-pro.com. Use the Locator+ codes shown below to download the code. Free online registration is required. See the Online Resources box in "Letters to the Editor" for more information.

An Introduction to XML for Java Programmers
Registered Locator+: **JP0399**
Premier Locator+: **JP0399PM**

Attention Developers

Search Articles. Download Code.



PREMIER Club

Join now and Premier Club membership is ONLY $20 for six months. You'll get an additional 25% off if you already subscribe to *VBPJ*, *WEB Builder* or *Java™ Pro*.

The Development Exchange's (DevX's) Premier Club offers you access to the largest resource of programming information and tools available anywhere. Your Premier Club membership starts with immediate, unlimited access to information and downloadable code from the entire archive of FTP's leading programming titles—information that will save you tons of coding time.

And as a Premier Club member, your access extends to any of the unique development zones and sites within the DevX family of web sites—all with related articles, key tools, and dedicated programming information on VB, C++, Java, and Enterprise development.

To start your membership, point your browser to DevX's Premier Club area, and follow the instructions.

Member Benefits

- **Access to All Previous Issues.** Over 1,000 articles from *Visual Basic Programmer's Journal*, *WEB Builder*, Avatar, *Java™ Pro*, and *Getting Started with Visual Basic*.
- **Code!** Members gain admission to the entire archive area of DevX, with complete searchable text from all these publications.
- **Even More Code.** Extra apps with complete source, plus special utilities.
- **Email Product Reviews.** Get e-mail issues of Visual Programming++ (VP++), with product reviews of the latest development tools as they are released.
- **Tools and Links.** Unlimited access to DevX's searchable product database, the only comprehensive, online listing of every significant programming tool on the market.
- **Club Privileges.** Special discounts, discussions, services, and much more.

Enroll Today at

www.premier-club.com

DEV·X The Development Exchange

www.devx.com

Microsoft, Windows, Visual Basic, Visual C++ are registered trademarks of Microsoft Corporation. Java is a registered trademark of Sun Microsystems.
The Development Exchange, DevX, CareerLink, and Ask the Pros are trademarks of Fawcette Technical Publications.
Fawcette Technical Publications, 209 Hamilton Avenue, Palo Alto, CA 94301-2500. Phone: 650.833.7100; Fax: 650.853.0230; www.devx.com.

Java Gets the Job Done

Despite some remaining deficiencies, Java is the tool of choice for an increasing number of developers

by Daniel F. Savarese

Daniel Savarese is a Senior Scientist at the Center for Advanced Computing Research at the California Institute of Technology. Daniel is the author of several Java packages, including the OROMatcher and PerlTools regular expression libraries, and the NetComponents network protocol library. Reach him at dfs@cacr.caltech.edu.

I find that I'm complaining less about Java's poor performance these days. Certainly, when I run benchmarks against C and C++, the performance disparity persists. But when I write user applications, response times are good and reliability is on par with native apps. I've talked to many developers who echo these sentiments. Java has often been accused of being a hype machine, and its developers of being religious zealots, but the reality is that most of us have been rather practical about the language, adopting it for appropriate tasks. From the beginning, we realized it was slow, but we also recognized the potential for performance improvement. The application-development benefits, combined with features such as mobile code and a built-in security model, were sufficient to warrant early adoption. Since then, a wealth of standard APIs have emerged, and the improvements in JIT technology continued, further increasing the value of our Java investments.

But it hasn't all been pretty. We've faced incompatibilities between various VMs. Applets lost much of their initial attraction due to the increased power of DHTML applications and the inability to reliably leverage the Swing classes. The ongoing dispute between Sun and Microsoft has cast a shadow of uncertainty over developers committed to deploying Java applications on the Win32 platform. However, few of these issues have had any impact on server-side Java, which has seen increased popularity since the introduction of the Servlets API.

So what do we still need to be concerned about when considering the deployment of Java-based solutions? Running multiple Java applications on a PC can be resource intensive. Even though processor and memory speeds are up, a Java application can easily consume a large percentage of these resources. It is still not terribly practical to run multiple Java applications on a user's desktop. However, this doesn't pose quite as large a problem on the server, where you do not use Java's GUI components. As a general rule, you will probably want to double the amount of RAM on your server if you plan on migrating your existing server apps to Java. However, Novell has done an excellent job of integrating its JVM with NetWare, making it possible to efficiently run many Java programs without requiring extra memory.

I/O still sucks. Even though input and output streams and readers are nice abstractions, in practice, they're not very efficient for I/O intensive applications. This is because much of their code is 100% Pure Java. We need an API that cuts through some of these layers and eliminates the overhead. Don't be afraid to write a little native code to spruce up your I/O performance if you need it.

Thread behavior is still poorly specified. Unfortunately, writing multithreaded applications requires more testing than it should. Thread priorities are only scheduling hints—you can't count on them to be interpreted consistently across platforms. If you are writing multithreaded servers, you are probably going to deploy them on a specific platform. If that isn't the case, it helps to write your code as though you're working in a non-preemptive thread scheduling environment. Insert yields at strategic locations, and use polling variables rather than the deprecated thread methods for suspending and killing threads.

Performance on Linux is lagging. Maybe you don't think this matters, but the way I see it, Linux is the only operating system (other than the Windows platform) that is increasing in market share. It is now clear that corporations will be adopting Linux in greater numbers for deploying

Listing 1 A Conversion Experience This program enables you to convert the contents of a frame into an Image object.

```
import java.awt.*;
import java.awt.event.*;

public class FrameToImage {

   static class ImageFrame extends Frame {
      private Image __image = null;

      public void paint(Graphics g) {
         if(__image == null)
   super.paint(g);
         else
   g.drawImage(__image, 0, 0, this);
      }

      public void setImage(Image image) { __image = image;
}
   }


   public static final void main(String[] args) {
      final Frame frame1;
      final ImageFrame frame2;
      final Button button;
      WindowListener exitListener;

      frame1 = new Frame();
      frame1.setSize(200, 200);
      frame1.setResizable(false);

      frame2 = new ImageFrame();
      frame2.setSize(200, 200);
      frame2.setResizable(false);

      exitListener = new WindowAdapter() {
         public void windowClosing(WindowEvent e) {
            Window window = e.getWindow();
            window.setVisible(false);
            window.dispose();
            System.exit(0);
         }
      };

      frame2.addWindowListener(exitListener);

      button = new Button("Convert to Image");
      button.addActionListener(new ActionListener() {
         public void actionPerformed(ActionEvent e) {
   Image image;
   Dimension size;

   size  = frame1.getSize();
   image = frame1.createImage(size.width, size.height);

   frame1.setVisible(false);
   frame1.printComponents(image.getGraphics());
   frame2.setImage(image);
   frame2.repaint();
      }
   });


      frame1.add(new Label("Frame 1", Label.CENTER),
         BorderLayout.CENTER);
      frame1.add(button, BorderLayout.NORTH);
      frame1.setLocation(10, 10);
      frame2.setBackground(Color.blue);
      frame2.setLocation(300, 10);
      frame1.setVisible(true);
      frame2.setVisible(true);
   }
}
```

mission-critical applications, especially in light of all the 1998 announcements from major database vendors. In case you hadn't heard, Informix, Oracle, DB2, Sybase, and just about any other major enterprise database system you can name that isn't produced by Microsoft will be ported to Linux and should be available when you read this. All of these databases make heavy use of Java as an application development language. But the port of the Sun JDK to Linux is a strictly interpreted JVM, and only with version 1.1.7A did it support native threads (previously it used the non-preemptive user space Green threads library). And even though several open source Just In Time (JIT) compiler projects support Linux, none are without bugs, and they don't approach the performance of commercially produced JITs available for the Win32, Solaris, AIX, HP-UX, and NetWare. Linux is such a critical platform that I have no doubt that the current problem will eventually be solved, but the simplest solution would be for Sun to officially support Linux. This would not require much effort and would generate many more Java sales for Sun. We'll see what happens.

Before I get to your questions, I want to mention that the Java 2 platform slipped in a new keyword, `strictfp`. The floating-point extensions' public review I mentioned in my last column has been concluded, and this new keyword is the result. The keyword is used to indicate that traditional Java floating-point behavior should be used (meaning that of versions prior to 1.1). The default floating point behavior has been changed so that it takes advantage of extra precision that might not be available on the native platform. Watch out, your floating-point code may now produce different (possibly more correct) results than it used to.

Q Is there a version of the JDK that will run on DOS instead of a Windows 95 machine?

A Sun has developed a product called JavaPC, an offshoot of JavaOS (currently in its fifth beta release) that is specifically designed to run on DOS. But there's a catch. You need at least 10 MB of disk space and 8 MB of RAM to run applets and small applications. Sun recommends 16 MB of RAM for running anything larger than that. However, most DOS machines never had that much RAM. My dusty, old 386/SX couldn't even be expanded beyond 8 MB of RAM. But there's more! JavaPC won't run on a 386 processor because it doesn't have an on-chip floating-point unit. According to the JavaPC reference guide, you need at least a 486 DX-2 processor to use JavaPC. But a 486 can handle Windows 95 pretty well and Linux even better, both of which support Java. So JavaPC is a tough sell. The primary market for it appears to be OEM's who want to build cheap network computers using 486 processors running DOS. You can learn more about JavaPC at www.javasoft.com/products/javapc/index.html.

Java has often been accused of being a hype machine

Q I have a list of filenames in a String array that I'd like to sort in alphabetical order. Is there a class to do this, or do I have to implement my own sort?

A If you're using JDK 1.2, you're in luck—you won't have to roll your own code. The `java.util.Arrays` class, in conjunction with the Collections API, provides a ready-made way of sorting arrays. Keep in mind that the sorts in the `Arrays` class are designed to work well with a large range of data sets, but might not work well with your specific data distribution. But for sorting a bunch of filenames, the provided sorting algorithms will be more than adequate. Using JDK 1.2, you could sort an array of filenames as simply as this:

```
String[] stringArray;

   java.util.Arrays.sort(stringArray);
```

But if you are not using JDK 1.2 yet, you will have to roll your own sort or use a third-party library such as ObjectSpace's JGL. Sorting is a well-studied area, and you can find many algorithms to choose from by visiting your local library or bookstore.

Q I want to convert the content of a component into a GIF image. I know how to convert an Image object to a GIF, but I don't know how to convert the contents of a frame into an Image object.

A Capturing the graphical contents of a container in an Image is done in much the same way as you would print the contents, except that instead of printing to a PrintGraphics object, you print to the Graphics object of the Image. The `printComponents()` method does all the work for you. The program in Listing 1 creates two frames, one containing a button and a label, and the other simply a blue background. When you press the button in the first frame, the program creates an offscreen image, prints the first frame to image, hides the frame, and paints the image to the second frame.

Q I need to call a constructor of a dynamically loaded class, and that constructor takes a single parameter that is an array of strings(such as `constructor(String [] args)`. Can I do this via reflection?

A Reflection can be a tiresome business, involving some arcane method calls and lots of exception catching. Getting a constructor via reflection is similar to getting a method. Once you have your class reference, you have to set up the parameter signature of the constructor by constructing an array containing references to the class types of the parameters. This can be done rather easily by using `Class.forName(String classname)`. The only catch is that array parameters require a funky naming format that is not documented in the API docs, but can be discovered through experimentation or looking at the Java Language Specification. Array class names are represented by a sequence of left brackets ([ ), one for each dimension of the array, then a capital L followed by the class name of type stored by the array, ended with a semicolon. For example, a `String` array is referenced as `"[Ljava.lang.String;."` Once you know this, life becomes simpler. Just use `Class.getConstructor` to fetch the con-

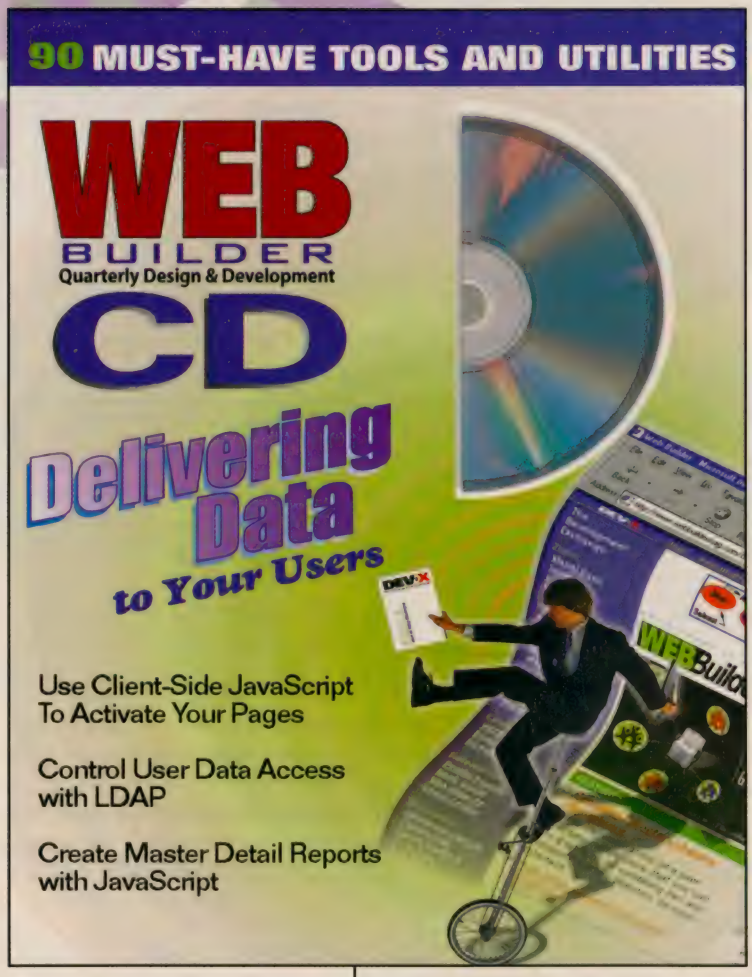
Listing 2 Calling a Constructor via Reflection Getting a constructor via reflection is similar to getting a method. This code demonstrates the steps and dynamically invokes a method that prints the values of the parameter passed to the constructor.

```
public final class ConstructorReflection {
  private String[] __strArray;

  public ConstructorReflection(String[] strArray) {
    __strArray = strArray;
  }

  public void print() {
    for(int i=0; i < __strArray.length; i++)
      System.out.println(__strArray[i]);
  }

  public static final void main(String[] args) {
    Constructor constructor;
    Method method;
    Class[] parameterTypes = new Class[1];
    Object[] parameters;
    Class clss;
    Object obj;

    try {
      // Get the class corresponding to
      // java.lang.String[],
      // which is the parameter type for the
      // constructor.
      parameterTypes[0] =
        Class.forName("[Ljava.lang.String;");
    } catch(ClassNotFoundException e) {
      e.printStackTrace();
      return;
    }

    clss = ConstructorReflection.class;

    try {
      constructor =
        clss.getConstructor(parameterTypes);
    } catch(NoSuchMethodException e) {
      e.printStackTrace();
      return;
    }

    // Set up the constructor parameters before creating
    // a new instance
    parameters    = new Object[1];
    parameters[0] = new String[] { "One", "Two",
      "Three", "Four" };

    try {
      obj = constructor.newInstance(parameters);
    } catch(InstantiationException e) {
      e.printStackTrace();
      return;
    } catch(IllegalAccessException f) {
      f.printStackTrace();
      return;
    } catch(InvocationTargetException g) {
      g.printStackTrace();
      return;
    }

      try {
      method = obj.getClass().getMethod("print", null);
    } catch(NoSuchMethodException e) {
      e.printStackTrace();
      return;
    }

    try {
      method.invoke(obj, null);
    } catch(IllegalAccessException e) {
      e.printStackTrace();
      return;
    } catch(InvocationTargetException f) {
      f.printStackTrace();
      return;
    }
  }
}
```

Build a better, smarter Web site

Only $29.95 for four quarterly issues—you'll save 42% off the newstand price!

With *WEB Builder CD*, your complete Web Development tool kit



All the Web development resources you need to take your site to the next level! Every issues delivers:

- Critical how-to articles from *VBPJ, Java™Pro, WEB Builder, Getting Started with Visual Basic,* and *Enterprise Development* magazines
- The latest versions of new tools and utilities
- Dozens of updated product reviews and fresh links

Plus, get valuable resources to continue and expand learning: links, software, tips, basics, code and more!

Subscribe Today—4 issues for $29.95

Order online at www.devx.com
Or call 1-800-848-5523 International orders: 650-833-7100
Fax 650-321-3818

FTP FAWCETTE TECHNICAL PUBLICATIONS **209 Hamilton Ave., Palo Alto, CA 94301**

Visual Basic is a registered trademark of Microsoft Corporation. Java is a trademark of Sun Microsystems. *VBPJ* is a trademark of Fawcette Technical Publications.

Listing 3 Static in an Inner Class Static member variables and methods are disallowed by the Inner Classes specification that was introduced with Java 1.1. However, by defining the static member as a member of the enclosing class, the static inner class can have static members. This example differentiates the behavior of inner classes and static nested classes with respect to static members.

```
public class InnerClassExample {
  private String __privateMember = "__privateMember";
  private static String __staticPrivateMember =
    "__staticPrivateMember";

  // This is an inner class, it cannot have static
  // members.
  public class InnerClass1 {
    public void print() {
      System.out.println(__privateMember);
      System.out.println(InnerClass2.__
        staticInnerClass2Member);
    }
  }


  // This is a static nested class, which makes it a
  // top-level class, allowing it to have static members.
  public static class InnerClass2 {
    public static String __staticInnerClass2Member =
      "__staticInnerClass2Member";

    public void print() {
      // Accessing __privateMember would be a
      // compile-time error. InnerClass2 is a
      // top-level class and can only
      // reference its own member variables
      // and the static variables of its
      // lexically encompassing class,
      // InnerClassExample.
      System.out.println(__staticPrivateMember);
      System.out.println(__staticInnerClass2Member);
    }
  }


  public void testInnerClasses() {
    InnerClass1 class1 = new InnerClass1();
    InnerClass2 class2 = new InnerClass2();

    class1.print();
    class2.print();
  }


  public static final void main(String[] args) {
    InnerClassExample example = new InnerClassExample();
    example.testInnerClasses();
  }
}
```

structor reference, using your parameter type array as an argument. New object instances can be created by invoking the contructors `newInstance()` method, passing it an array of parameters expected by the constructor. Listing 2 demonstrates these steps and dynamically invokes a method that prints the values of the parameter passed to the constructor.

Q How can a unique filename be constructed to guarantee that it will be unique across processes?

A There is unfortunately no Java 1.1 analog to the POSIX `tmpnam()` C function, which generates a unique filename, or `tmpfile()` and creates a new file with that filename. But Java 2 has added this functionality with the `createTempFile()` static method to the `java.io.File` class. This method will create a new file with a unique name based upon a base pattern. During any given run of an application, `createTempFile` will never create the same filename twice. If, for some reason, it cannot create a file that did not previously exist, `createTempFile()` will throw an `IOException`. I cannot conceive of a bulletproof way of implementing this functionality in Java 1.1 using 100% Pure Java code. In fact, Java 2's `createTempFile()` relies on native code to implement the method.

Figure 1 Creating a Splashy Screen Splash screens keep your user's screen from going blank while your application loads and can provide valuable information. I used JDialog for this splash page to keep it top of the parent window and to prevent the native window manager from adorning it with a border.

Q Why can't I have a static member in an inner class?

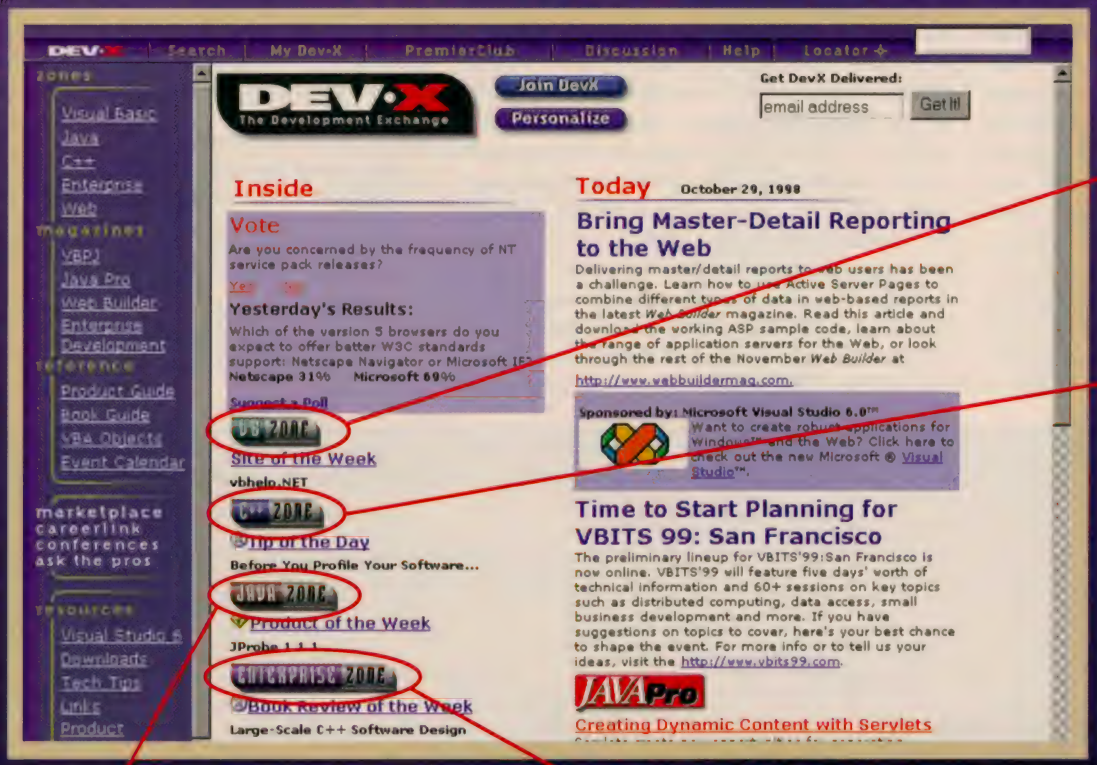
A The simple answer is that static member variables and methods are disallowed by the Inner Classes specification that was introduced with Java 1.1. Why is this disallowed by the specification? First, I should distinguish between an inner class and a static inner class. An inner class is a class declared within the scope of another class, either local to a class declaration (such as a class member), a block of statements, or an expression (in which case it is an anonymous class). However, it is possible to define a static inner class within a top-level class declaration. A class defined in this way is treated as a top-level class, meaning that it can only access its own members and static members of its enclosing class. But because it is declared within the scope of another class, it cannot be referenced outside of the enclosing class without specifying a qualified name (such as `OuterClassName.InnerClassName`).

Inner classes cannot have static members because static members are top-level lexical entities. In other words, static members are accessible outside of a given class' scope. Inner classes themselves are not top-level entities, being bound to specific instances of their enclosing class. If you feel you really need an inner class to have static member, just define that variable as a member of the enclosing class. But the static inner class can have static members. This is because they are top-level classes, and as such are not bound to specific instances of their enclosing class. The best way to think about this is to consider inner classes and static inner classes as two different animals by introducing the term "nested class," which is loosely used in the Inner Classes specification. Think of both types of classes as nested classes, and only use the term "inner class" to refer to non-static nested classes. Therefore a static nested class is not really an inner class and can have static members. The example in Listing 3 differentiates the behavior of inner classes and static nested classes with respect to static members.

Q How do I show a splash page that would time out after approximately five seconds?

A Presumably you would like to display a splash page while your application is loading,

www.devx.com


DEV·X
The Development Exchange
Join DevX
Personalize
Get DevX Delivered:
email address
Get It!
Inside
Vote
Are you concerned by the frequency of NT service pack releases?
Yesterday's Results:
Which of the version 5 browsers do you expect to offer better W3C standards support: Netscape Navigator or Microsoft IE?
Netscape 31%
Microsoft 69%
Site of the Week
vbhelp.NET
Tip of the Day
Before You Profile Your Software...
Product of the Week
JProbe 1.1.1
Book Review of the Week
Large-Scale C++ Software Design
Today
October 29, 1998
Bring Master-Detail Reporting to the Web
Delivering master/detail reports to Web users has been a challenge. Learn how to use Active Server Pages to combine different types of data in Web-based reports in the latest Web Builder magazine. Read this article and download the working ASP sample code, learn about the range of application servers for the Web, or look through the rest of the November Web Builder at
http://www.vbbuildermag.com.
Sponsored by: Microsoft Visual Studio 6.0™
Want to create robust applications for Windows™ and the Web? Click here to check out the new Microsoft ® Visual Studio™.
Time to Start Planning for VBITS 99: San Francisco
The preliminary lineup for VBITS'99:San Francisco is now online. VBITS'99 will feature five days' worth of technical information and 60+ sessions on key topics such as distributed computing, data access, small business development and more. If you have suggestions on topics to cover, here's your best chance to shape the event. For more info or to tell us your ideas, visit the http://www.vbits99.com.
JAVAPro
Creating Dynamic Content with Servlets
zones
Visual Basic
Java
C++
Enterprise
Web
magazines
VBPJ
Java Pro
Web Builder
Enterprise Development
reference
Product Guide
Book Guide
VBA Objects
Event Calendar
marketplace
careerlink
conferences
ask the pros
resources
Visual Studio 6
Downloads
Tech Tips
Links
Product

The VB Zone:
www.vb-zone.com

The C++ Zone:
www.cplus-zone.com

Each DevX Zone is a Web site, targeting your specific information needs for techniques and tools, from VB to Java, from C++ to SQL.

The Java Zone:
www.java-zone.com

The Enterprise Zone:
www.enterprise-zone.com

Need programming information, NOW?

It's all here on The Development Exchange (DevX)

DevX's family of Web sites is your most comprehensive source for programming information available. Get immediate access to essential tools, code, and resources aimed at helping you solve your toughest programming challenges. And now, whether you're undertaking intranet solutions with Java, end-user applications with VB, or ISV products with C++, **you're just a click away from the right tool in one of DevX's four development zones.**

- **Technical articles and code** from *Visual Basic Programmer's Journal*, *Enterprise Development*, and *Java™Pro* magazines
- **Daily programming tips**—plus a library of thousands more
- **Ask the Pros**—get answers to tough development questions
- **Product Reviews**—hundreds of reviews updated weekly
- **Forums**—dedicated to development
- **40% off Books** of the Week—plus editors' picks and reviews of top programming titles
- **Search Products**—from our database of over 5,000 programming tools
- **Conference updates**—register online

And that's just the beginning.

DEV·X
The Development Exchange

Explore the rest yourself at
www.devx.com

Microsoft, Windows, Visual Basic, Visual C++ are registered trademarks of Microsoft Corporation. Java is a registered trademark of Sun Microsystems.
The Development Exchange, DevX, CareerLink, and Ask the Pros are trademarks of Fawcette Technical Publications.
Fawcette Technical Publications, 209 Hamilton Avenue, Palo Alto, CA 94301-2500 Phone: 650.833.7100 Fax: 650.853.0230 www.devx.com

rather than freezing the application for five seconds. Any time you start talking about doing two things at once, you've got to think threads. You could always display the splash page only while the application is loading, in which case you would not need to use threads. Just show the splash page, load the application, and hide the splash page. But many applications load very quickly, and you would often like the user to see your splash page for a specified period of time.

The `Thread.sleep()` method allows you to introduce a delay into your program, suspending the current thread of execution for the specified amount of time before resuming. There are two versions of `Thread.sleep()`. The first accepts a single long argument that specifies the delay in milliseconds. The second accepts an additional integer argument that specifies an additional number of nanoseconds. Now that you know how to make a thread wait for a specified period of time, all you have to do is create a separate thread that shows the splash page, sleeps for five seconds (5,000 milliseconds), and hides the splash page after waking up. Listing 4 demonstrates how to do this using the Swing classes. I use a `JDialog` for the splash page because the native window manager will adorn it with a border and keep it placed on top of the parent window. `JWindow` instances have no borders and cannot take advantage of the native window system to keep them placed on top of a frame. I've provided a sample splash page (see Figure 1) which is available on the *Java Pro* Web site. Notice how I have to calculate the coordinates of the splash page in order to center it on the application window. ■

Code Online

All the code in this article is available for free at www.java-pro.com. Use the Locator+ codes shown below to download the code. Free online registration is required. See the Online Resources box in "Letters to the Editor" for more information.

Java Gets the Job Done
Registered Locator+: **JP0399**
Premier Locator+: **JP0399AP**

Listing 4 Doing Two Things at Once If you want to load your application and display a splash page, you need to use threads. This example creates a separate thread and displays the splash page using the Swing classes. The thread sleeps for five seconds and then hides the splash page after waking up.

```
import java.awt.*;
import java.awt.event.*;
import com.sun.java.swing.*;


public class SplashPage {
  JFrame __frame;

  static class SplashPanel extends JPanel {
    private Icon __splashIcon;

    SplashPanel(Icon icon) {
      __splashIcon = icon;
      setPreferredSize(new
        Dimension(icon.getIconWidth(),
            icon.getIconHeight()));
    }

    public void paint(Graphics graphics) {
      super.paint(graphics);
      __splashIcon.paintIcon(this, graphics, 0, 0);
    }
  }


  class SplashThread extends Thread {
    private int __delay;

    SplashThread(int delay) { __delay = delay; }

    // Displays splash page on top of application frame,
    // waits __delay milliseconds, and disposes of the
    // splash page.
    public void run() {
      SplashPanel panel = new SplashPanel(new
        ImageIcon("splash.gif"));
      JDialog splash;
      Dimension size1, size2;
      Point location;

      location = __frame.getLocation();
      size1    = __frame.getSize();
      splash   = new JDialog(__frame);
      splash.getContentPane().add(panel);
      splash.pack();
      size2     = splash.getSize();
      splash.setLocation(location.x +
        (size1.width - size2.width)/2,
      location.y + (size1.height - size2.height)/2);
      splash.setVisible(true);
      // Wait five seconds, then dispose of splash.
      try {
  Thread.sleep(__delay);
      } catch(InterruptedException e) {
  // Don't do anything
      }
      splash.setVisible(false);
      splash.dispose();
    }
  }


  public SplashPage() {
    WindowListener exitListener;

    __frame = new JFrame("Splash Demo");
    exitListener = new WindowAdapter() {
      public void windowClosing(WindowEvent e) {
        Window window = e.getWindow();
        window.setVisible(false);
        window.dispose();
        System.exit(0);
      }
      };
    __frame.addWindowListener(exitListener);

    // any other application initialization goes here
  }


  public void loadApplication() {
    Thread showSplash;
    Dimension size;

    size = Toolkit.getDefaultToolkit().getScreenSize();
    __frame.setSize(640, 480);

    // Center frame on screen.
    __frame.setLocation((size.width - 640)/2,
      (size.height - 480)/2);
    __frame.setVisible(true);

    // Show the splash page for 5 seconds
    showSplash = new SplashThread(5000);
    showSplash.start();

    // Continue loading your application here while the
    // splash page is being displayed.
  }


  public static final void main(String[] args) {
    SplashPage application;

    application = new SplashPage();
    application.loadApplication();
  }

}
```

Advertiser Index

Caribou Lake Software 4
1-612-921-8338
http://www.CaribouLake.com

Compuware, NuMega 25
1-800-4-NUMEGA
http://www.numega.com

DevX-C++-Zone 43
1-800-848-5523
http://www.Cplus-zone.com

DevX 59
1-800-848-5523
http://www.devx.com

Dev-Enterprise-Zone 41
1-800-848-5523
http://www.enterprise-zone.com

DevX-Java-Zone 47
1-800-848-5523
http://www.java-zone.com

DevX-Premier Club 53
1-800-848-5523
http://www.premier-club.com

Diab Data 78
http://www.ddi.com

Distinct Software C3
http://www.distinct.com

Enterprise Development 2000 51
1-800-848-5523
http://www.enterprisedev.com

Enterprise Development 65
Magazine Subscriptions
1-800-848-5523
http://www.devx.com

EnterpriseSoft 17
http://www.EnterpriseSoft.com

FTP Books 76
1-800-848-5523
http://www.devx.com

IBM Corporation 8
http://www.software.ibm.com/udb98

Inquiry.com 45
1-800-848-5523
http://www.inquiry.com

Intuitive Systems 6
http://www.optimizeit.com

Java Pro 33
Magazine Subscriptions
1-800-848-5523
http://www.devx.com

KL Group C2 & 1
1-800-663-4723
http://www.klg.com

ObjectShare 27
1-800-973-4777
http://www.objectshare.com

ObjectSoft, Inc. 31
1-888-OBJSOFT
http://www.objsoft.com

ProtoView Development Corporation C4
1-800-231-8588
http://www.protoview.com/java

RSA Data 37
1-800-782-5453
http://www.rsa.com/developersolutions

Sybex Computer Books 19
http://www.sybex.com

Symantec 11
1-888-822-3409
http://www.cafe.symantec.com

Tidestone Technologies 23
1-800-884-8665
http://www.tidestone.com

TGVP 69
1-800-848-5523
http://www.devx.com

Visual Numerics 13
1-800-364-8880
http://www.vni.com/javapro-march

Web Builder CD 57
1-800-848-5523
http://www.devx.com

ZeroG Software 2
1-415-512-7771
http://www.ZeroG.com

Showcase

Caribou Lake Software 61
1-612-921-8338
http://www.CaribouLake.com

Data Representations 61
1-888-307-9550
http://www.datarepresentations.com/

ErgoTech Systems Inc. 61
1-888-374-6832
http://www.ergotech.com

Product Information

- To get immediate details on products advertised in this issue, you can simply log onto our online services at **http://infoasap.devx.com**.
- This provides a list of direct links to this month's advertiser's Web sites.

Advertising Sales Contacts

- **Kent Wu, Advertising Director**
  650-833-7127; kwu@fawcette.com
- **Regan Swett, Regional Sales Manager**
  650-833-7154; rswett@fawcette.com
- **Kelly Keane, Western Regional Sales Manager**
  650-833-7128; kkeane@fawcette.com
- **Shana Deane, Eastern Regional Sales Manager**
  413-586-6074; sdeane@fawcette.com

This listing is provided as a courtesy to our readers and advertisers. The publisher assumes no responsibility for errors or omissions.

Sales Department: 209 Hamilton Avenue • Palo Alto, CA, USA 94301-2500 • 650-833-7100

Product Showcase

JAVAPro

www.java-zone.com

Java Solutions Delivered Daily

Jump on today and get immediate access to the most comprehensive source of expert information and code to help you develop serious Java applications, including:

- Technical articles and code
- Daily programming tips
- Download Java tools
- Search Products
- Product Reviews
- Dedicated Java Forums
- 40% off Books of the Week

DEV·X The Development Exchange

Fawcette Technical Publications
209 Hamilton Avenue • Palo Alto, CA 94301-2500
Phone: 650.833.7100 • Fax: 650.853.0230
www.devx.com

Java is a registered trademark of Sun Microsystems, Inc.
DevX is a trademark of Fawcette Technical Publications.

http://www.devx.com

JSQL™

FOR THOSE SERIOUS ABOUT JAVA DEVELOPMENT

Recently named to the Information Week's Top 100 Internet and Intranet Solution Series (10/5/98 Issue), JSQL™ provides the middleware you need to develop secure, robust, scalable, high performance bridges between Java and your Enterprise applications, databases and data warehouses.

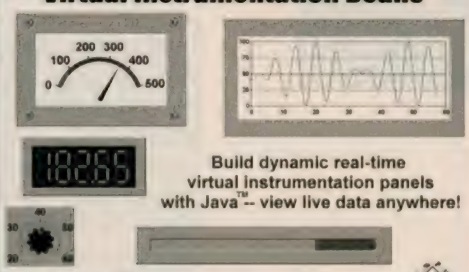
CARIBOU LAKE SOFTWARE

"Superior Java/RDBMS/DW Products and Services"

8400 Normandale Lake Blvd., #920
Minneapolis, MN 55437
Tel: 612 921-8330 Fax: 612 921-8338

http://www.CaribouLake.com

Virtual Instrumentation Beans™



Build dynamic real-time virtual instrumentation panels with Java™ -- view live data anywhere!

ErgoTech Systems, Inc.
(888) ERGOTECH (505) 662-5156

http://www.ergotech.com

FREE DOWNLOAD!

The Simplest Way to Create Java...

Simplicity for Java

- The Code Sourcerer™ writes Java code for you
- Executes any Java code on-the-fly without compiling
- Supports all AWT & Swing components
- Import any JavaBeans™

http://www.datarepresentations.com

OO Ideas and Plotting Algorithms

Creating good Java programs requires you to think in objects, but the results can be much simpler code

by James W. Cooper

The process of learning object-oriented (OO) programming requires a complete shift in mental gears, sometimes called a "paradigm shift." You have to start thinking about objects that contain both data and methods, rather than just arrays or structures. This is especially true when you write programs in Java—you absolutely must write all of your programs using objects.

Recently, a swim coach asked me to devise a program to plot swimmers' progress over time, allowing them to compare several strokes or distances at once. As I began to consider the design for this program, I realized that it was an ideal example of how object-oriented programming actually simplifies an otherwise complicated project. The simple example of plotting multiple curves that I outline in Figure 1 would be much more difficult to write if we didn't use OO techniques to encapsulate some of the information.

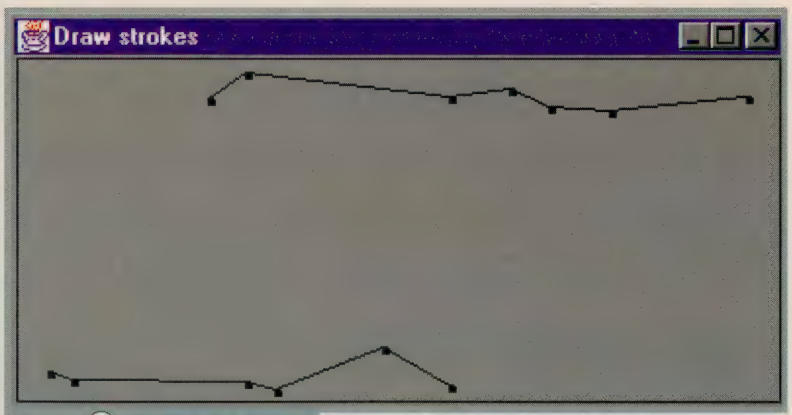


Figure 1 Plotting the Curve This graph plots a swimmer's time improvements in the 100- and 200-yard freestyle events.

The Plot Outline

Now let's consider the problem we want to solve. A coach provides us with a set of times for one or more swimmers and we need to plot these times versus the dates they were achieved. He would also like to see several different strokes or several distances of the same stroke on the same chart, to compare improvements. In general swimmers do not swim all strokes and all distances at every meet, so the number of data points for each stroke/distance will differ as well as the dates when these times were achieved.

Although we'd prefer to use a standard plotting Bean to plot these data, there are a couple of reasons we might choose to draw the lines ourselves inside a panel. First, it is difficult to find good, inexpensive plotting Beans, and second, most plotting algorithms in standard packages (like Excel) require that one or both of the axes be integral, rather than both being values that can vary continuously. In this case, both the dates and the times are effectively floating-point single precision numbers. We need to plot a variable number of arrays, and each array has an unpredictable number of data points.

Clearly it is difficult to represent these data as two-dimensional arrays, because we don't know how many of them there will be or for how long. Although we could use a series of arrays to represent these data and redimension them with new to change their size as needed, this can become very complex to track.

The Plot Thickens

Rather than using arrays, let's consider using objects to represent these data and encapsulate this unpredictability.

James W. Cooper is the author of twelve books on computing and is working on a book on Java design patterns for Addison-Wesley. He also writes swimming software in his spare time.

100-yd Freestyle		200-yd Freestyle	
Time	Date	Time	Date
103.66	10-28-95	216.80	11-02-95
101.53	11-02-95	209.18	12-01-95
100.95	12-09-95	215.27	12-09-95
59.37	12-15-95	209.65	1-20-96
109.49	1-06-96	211.45	1-01-96
100.23	1-20-96	207.47	2-10-96
		206.11	2-23-96
		209.87	3-23-96

Table 1 Typical Swimmer Data This is the information that would be used by our application to plot swimmer's times on a graph to allow comparison, tracking their progress.

Using objects in this manner can turn a potential jungle of arrays into an incredibly simple, elegant little program that is easy to understand and easy to maintain.

Remember that an object in OO programming is a combination of a structure that holds data and the subroutines or methods that can operate on that data. Just as important, you can have any number of instances of an object, with each holding different data, but utilizing the same methods. We'll use three objects in this program: A time-date pair object, a time-date pair vector, and a plotting display object.

Each object knows about its data and how it must be handled and converted to a convenient format. For example, Table 1 shows the form of the date and time datapoint pairs in our input file. The minutes and seconds are represented in these datafiles *without* the intervening colon. However, whether the colon is present or not, we need to convert the data to seconds in order to plot it linearly. Likewise, we need to convert the date values into some sort of linear form. These conversions are best done inside a simple `DataPoint` object that stores the data and returns the values in a convenient linear form, as shown in Listing 1.

Dates in Java

Dates in Java are best handled using the `GregorianCalendar` class, because most of the Date class methods have been deprecated. To convert data from a string like "1-20-96" to an actual date representation, we use the SimpleDateFormat class. Because we want to return dates as fractions along a year timeline axis, we create a floating-point representation where the year is the integer and the fraction is the number of 366ths of the way through the year. This is shown in the `floatDate` class:

```
public class floatDate extends Date  {
  GregorianCalendar date;
  //--------------------------------
  public floatDate(String dat)    {
    date = new GregorianCalendar();
    SimpleDateFormat fmt = new
      SimpleDateFormat("MM-dd-yyyy");
    date.setTime(fmt.parse(dat,
      new ParsePosition(0)));
    }
  //--------------------------------
  public float getFloatDate()    {
  float day =
    date.get(Calendar.DAY_OF_YEAR);
  float year = date.get(Calendar.YEAR) +
    (day/366.0f);
  return year;
  }
}
```

The StrokeSet Object

In the more-complex `StrokeSet` object, each instance contains one array of `DataPoint` objects. Because the array may be of any size, we implement the array using the Vector class. But rather than just using a vector alone, we create the `StrokeSet` object, which keeps track of the maximum and minimum values of the time and date axes as we add new points into the object. Our calling routine creates an instance of the `StrokeSet` object and allows it to read from a data file. It then creates an instance of a `DataPoint` object for each line it reads from the file (see Listing 2). As we add each `DataPoint` object into the `StrokeSet` collection, it recalculates the x and y maxima and minima. Note that we initialize the max and min variables when we create each instance of the class so that we are comparing the array values to something meaningful.

There's more to OO programming than just creating a couple of classes

Now you can see the crux of this OO design: Each instance of the `StrokeSet` object contains one array of `DataPoints`. Each array can be of a different size and we can get the maxima and minima of each one by asking each `StrokeSet` instance. We can get the size of each array from the object, and we can get the values of each `DataPoint` from the object using the enumeration meth-

Listing 1 Converting Time The DataPoint object stores the time data and returns that value in a convenient linear form.

```
public class DataPoint {
  private float time;
  private floatDate date;

  public DataPoint(float times, String dat)   {
    time = makeSeconds(times);
    date = new floatDate(dat);
  }
//--------------------------------
private float makeSeconds(float t) {
  int mins;
  float secs;
  //convert time to seconds and store it
  mins = (int)t / 100;
  secs = t - (mins * 100);
  return mins * 60 + secs;
}
//--------------------------------
public float getTime() {
  return time;      //return time in seconds
}
//--------------------------------
public float getDate() {
  return date.getFloatDate();
 }
}
```

Listing 2 Set Up a Date To get our dates in the correct format our calling routine creates an instance of the StrokeSet object and allows it to read from a data file. It then creates an instance of a DataPoint object for each line it reads from the file.

```
public class StrokeSet {
  private Vector dataList;
  private float minTime, maxTime;
  private float minDate, maxDate;
  //-----------------------------------------
  public StrokeSet()   {
  SimpleDateFormat  fmt = new SimpleDateFormat();
  dataList = new Vector();
  minTime = 10000;
  maxTime = 0;
  minDate =  new floatDate("01-01-2100").getFloatDate();
  maxDate = new floatDate("01-01-1900").getFloatDate();
  }
  //-----------------------------------------
  public void readFile(String file)   {
    String s = "";
    InputFile f = new InputFile(file);
    s = f.readLine();
    while(s != null)      {
      int i = s.indexOf(",");
      if (i > 0)        {
      String dt = s.substring(0, i);
      float time = new
        Float(s.substring(i+1).trim()).floatValue();
      addPoint(time, dt);
      }
      s = f.readLine();
    }
    f.close();
  }


  public void addPoint(float times, String dat)   {
    DataPoint dp;
    dp = new DataPoint(times, dat);
    dataList.addElement(dp);

    //recalculate mins and maxes
    if (dp.getTime() > maxTime) maxTime = dp.getTime();
    if (dp.getTime() < minTime) minTime = dp.getTime();
    if (dp.getDate() < minDate) minDate = dp.getDate();
  if (dp.getDate() > maxDate) maxDate = dp.getDate();

  }
```

Listing 3 Displaying Our Data Our final object is the form that displays the plotted data. We create a StrokePlot object, which will plot data from a set of StrokeSet objects. It has an add() method, which recalculates the overall scale of the plot whenever we add in another StrokeSet.

```
public void add(StrokeSet st)   {
    if (xmin > st.getMinDate()) xmin = st.getMinDate();
    if (xmax < st.getMaxDate()) xmax = st.getMaxDate();
    if (ymin > st.getMinTime()) ymin = st.getMinTime();
    if (ymax < st.getMaxTime()) ymax = st.getMaxTime();
    strokes.addElement(st);
    rescale();
  }

private void rescale()   {
    Dimension sz = getSize();
    height = sz.height;
    width = sz.width;
    calcXscale();
    calcYscale();
  }
  //-------------------------------------
  private void calcXscale()   {
    xoffset = xmin-(0.05f*(xmax-xmin));
    xscale = 1.1f * (xmax - xmin)/ width;
  }
  //-------------------------------------
  private void calcYscale()   {
    yoffset = ymin- (0.05f *(ymax-ymin));
    yscale = 1.1f * (ymax - ymin)/ height;
  }
  And finally, our plotting takes place in a simple loop:

private int calcX(float x)   {
 return (int)((x - xoffset)/xscale);
}
//-------------------------------------
private int calcY(float y)   {
  return height - (int)((y - yoffset)/yscale);
}
//-------------------------------------
public void paint(Graphics g)
  {
    int x, y;

    g.drawRect(1, 1, width-2, height-2);
    for(int s=0; s < strokes.size(); s++)
    {
      StrokeSet st = (StrokeSet)strokes.elementAt(s);
      Enumeration e = st.elements();
      DataPoint dp = (DataPoint)e.nextElement();
      int oldx = calcX(dp.getDate());
      int oldy = calcY(dp.getTime());

      while(e.hasMoreElements())
      {
        dp = (DataPoint)e.nextElement();
        x = calcX(dp.getDate());
        y = calcY(dp.getTime());
        g.drawLine(oldx, oldy, x, y);
        g.fillArc(oldx, oldy, 4,4,0, 360);
        oldx = x;
        oldy = y;
      }
      g.fillArc(oldx, oldy, 4,4,0, 360);
    }

  }
}
```

ods of the Vector object. The advantage is that they can all be different sizes (or even empty!) without any real modification to our design.

Our final object is the form that displays the plotted data. We create a `StrokePlot` object, which will plot data from a set of `StrokeSet` objects. It has an `add()` method, which recalculates the overall scale of the plot whenever we add in another `StrokeSet` (see Listing 3). Our plotting takes place in a simple loop. Plotting axis tick marks and labels is also the responsibility of this `StrokePlot` object, but it is beyond the scope of this example. We'll take up some of the details of date handling and axis manipulation in a future column.

You've seen here that by putting a Vector inside another object you have all of the benefits of OO programming. Rather than worrying about separately named arrays or Vectors for each stroke data set, you can create new instances of the `StrokeSet` object and keep one stroke in each instance.

There's more to OO programming than just creating a couple of classes that we call from `main()`. In fact, creating objects that contain other objects is frequently the key to simpler and easier to understand programs. That is the real shift in thinking that will make you a better Java programmer. ■

Code Online

All the code in this article is available for free at www.java-pro.com. Use the Locator+ codes shown below to download the code. Free online registration is required. See the Online Resources box in "Letters to the Editor" for more information.

OO Ideas and Plotting Algorithms
Registered Locator+: **JP0399**
Premier Locator+: **JP0399JT**

RECEIVE YOUR FREE SUBSCRIPTION

to ENTERPRISE DEVELOPMENT—the essential tool for the management and architecture of application development projects.

I wish to receive ENTERPRISE DEVELOPMENT Magazine free of charge! ☐ Yes! ❑ No

NAME _____ TITLE _____

SIGNATURE (REQUIRED) X _____ COMPANY _____

DATE (REQUIRED) ____/____/____ DEPARTMENT _____

PHONE (____) _____ ADDRESS _____

FAX (____) _____ CITY, STATE _____

E-MAIL ADDRESS _____ ZIP/POSTAL CODE _____ COUNTRY _____

Annual subscription rate is $99.95 for nonqualified subscribers.

PLEASE ANSWER ALL QUESTIONS ON BOTH SIDES, SIGN AND DATE THE CARD. INCOMPLETE FORMS CANNOT BE PROCESSED.

1. PRIMARY BUSINESS ACTIVITY AT YOUR LOCATION (SELECT ONLY ONE):

01 ❑ Finance, banking, accounting, insurance
02 ❑ Manufacturing other than computer hardware and software
03 ❑ Manufacturing computers and computer software
04 ❑ Non-computer retailer, distributor, wholesaler
05 ❑ Applications software development
06 ❑ Aerospace
07 ❑ Government, military
08 ❑ Agriculture, mining, construction, oil & gas
09 ❑ Transportation, utilities
10 ❑ Media, marketing, advertising
11 ❑ Real estate
12 ❑ Research and development
13 ❑ Engineering
14 ❑ VAR/Systems Integrators
15 ❑ Communications services
16 ❑ Computer-related retailer, distributor, wholesaler
17 ❑ Systems software development
18 ❑ Education
19 ❑ Health, medical, legal
20 ❑ Consulting (please specify)_____
99 ❑ Other (please specify) _____

2. WHAT IS YOUR CURRENT JOB FUNCTION? (SELECT ONLY ONE FROM THE FOLLOWING SIX (6) CATEGORIES):

Application development manager/architect:
01 ❑ Applications development manager
02 ❑ Software development director
03 ❑ Software architect
04 ❑ Software project leader
05 ❑ Project manager
06 ❑ Advanced technology evaluation
07 ❑ Software team leader
08 ❑ Software engineering supervisor
09 ❑ Systems analyst/designer
10 ❑ Database designer
11 ❑ Application management/ administration
99 ❑ Other (please specify) _____

IS management:
30 ❑ CIO, Technical VP
31 ❑ IS/MIS manager
99 ❑ Other (please specify) _____

Application development staff:
40 ❑ Applications developer/programmer
99 ❑ Other (please specify) _____

Computer operations staff:
50 ❑ System Administrator
51 ❑ LAN Administrator
52 ❑ Help Desk
99 ❑ Other (please specify) _____

Corporate management:
60 ❑ CEO, president, partner, chairman, owner
61 ❑ VP, controller, director of marketing/sales, General manager, division/department head

Other (categories not listed above):
99 ❑ Other (please specify)

_____

3. ARE YOU DEVELOPING APPLICATIONS THAT INVOLVE WINDOWS NT OR DO YOU PLAN TO BE IN THE NEXT 12 MONTHS?

01 ❑ Yes 02 ❑ No

4. ARE YOU DEVELOPING APPLICATIONS THAT WILL INVOLVE ONE OR MORE SERVERS OR HOSTS OR DO YOU PLAN TO BE IN THE NEXT 12 MONTHS?

01 ❑ Yes 02 ❑ No

5. DO YOU EVALUATE, SPECIFY, RECOMMEND, AUTHORIZE, OR APPROVE THE PURCHASE OF (PLEASE CHECK ALL THAT APPLY):

01 ❑ Application development tools?
02 ❑ Database products?
03 ❑ Internet and/or Intranet products?
04 ❑ Middleware products?
05 ❑ Hardware servers?
06 ❑ Modeling tools
07 ❑ Testing/QA tools
08 ❑ Software process tools (Project/Process Management, Metrics, CM, etc.)
99 ❑ None of the above

6. WHICH OF THE FOLLOWING SOFTWARE SERVERS DO YOU BUY, SPECIFY, RECOMMEND, OR APPROVE THE PURCHASE OF AND/OR CURRENTLY HAVE INSTALLED AT THIS LOCATION (PLEASE CHECK ALL THAT APPLY):

	Buy, Specify, Recommend, Approve	Installed
Web	01 ❑	02 ❑
Database	03 ❑	04 ❑
Application	05 ❑	06 ❑
Transaction	07 ❑	08 ❑
E-mail	09 ❑	10 ❑
Imaging	11 ❑	12 ❑
Commerce	13 ❑	14 ❑
Remote Access	15 ❑	16 ❑
Communication/SNA	17 ❑	18 ❑

99 ❑ None of the above (over)

For faster service fax both sides of this form to: **1-650-321-3818**
OR fill out the electronic form at www.enterprisedev.com

7. SERVER/HOST OPERATING SYSTEMS IN USE AT THIS ENTIRE LOCATION (PLEASE CHECK ALL THAT APPLY):

01 ❑ Windows NT Server
02 ❑ Unix (all flavors)
03 ❑ Netware/Intranetware
04 ❑ OS/400
05 ❑ VMS
06 ❑ MVS/VM
07 ❑ OS/2
99 ❑ Other (please specify) _____

8. WHAT IS THE ANNUAL DOLLAR VALUE OF COMPUTER PRODUCTS AND SERVICES FOR WHICH YOU ARE PERSONALLY INVOLVED IN THE PURCHASING DECISIONS FOR YOUR ORGANIZATION'S OR LARGEST CLIENT'S INTERNAL USE?

01 ❑ More than $10 million
02 ❑ $5 million - $9,999,999
03 ❑ $2 million - $4,999,999
04 ❑ $1 million - $1,999,999
05 ❑ $500,000 - $999,999
06 ❑ $100,000 - $499,999
07 ❑ $50,000 - $99,999
08 ❑ $10,000 - $49,999
09 ❑ Less than $10,000

9. WHICH OF THE FOLLOWING CATEGORIES OF PRODUCTS DO YOU RECOMMEND, EVALUATE, SPECIFY, BUY, OR APPROVE THE PURCHASE OF? (CHECK ALL THAT APPLY):

01 ❑ Back office suites
02 ❑ N-Tier Capable Development Tools
03 ❑ ORBs
04 ❑ RPC middleware
05 ❑ Message Oriented Middleware
06 ❑ Application Integration Software
07 ❑ TP Monitors or Transaction Servers
08 ❑ Application Servers
09 ❑ Application Management Software
10 ❑ Component software (VBX, OCX, ActiveX, custom controls, JavaBeans, etc.)
11 ❑ Configuration Management/Version Control
12 ❑ Project Management Tools
13 ❑ Process Management Tools
14 ❑ Estimation/planning/metrics tools
15 ❑ Class Libraries
16 ❑ Application Frameworks
17 ❑ C/C++ Tools
18 ❑ Java Tools
19 ❑ VB Tools
20 ❑ 4GL Tools
21 ❑ OLAP, Data Mining or Data Warehousing tools
22 ❑ Reporting Tools
23 ❑ Web Authoring Tools
24 ❑ Expert systems/rules engines
25 ❑ Repository
26 ❑ Security Tools/Firewall Software
99 ❑ Other (please specify) _____
99 ❑ None of the above

10. DO YOU RECOMMEND, SPECIFY, APPROVE OR PURCHASE APPLICATION PACKAGES SUCH AS THOSE SOLD BY SAP, PEOPLESOFT, AND BAAN?

01 ❑ Yes 02 ❑ No

If YES, which types? (Check ALL that apply):

01 ❑ ERP Suites
02 ❑ Human Resources
03 ❑ Sales Automation
04 ❑ Financial/Accounting
05 ❑ Retail/Point of Purchase
06 ❑ Manufacturing/Inventory
07 ❑ Supply Chain Management
99 ❑ Other (please specify) _____

11. DO YOU RECOMMEND, SPECIFY, APPROVE OR PURCHASE APPLICATION DEVELOPMENT TOOLS FOR CUSTOMIZATION OF AND INTEGRATION WITH ERP OR OTHER APPLICATION PACKAGES?

01 ❑ Yes 02 ❑ No

12. WHAT IS THE NUMBER OF EMPLOYEES OF THE LARGEST BUSINESS ENTITY BY WHOM YOU ARE EMPLOYED?

01 ❑ Under 100
02 ❑ 100 - 999
03 ❑ 1,000 - 4,999
04 ❑ 5,000 - 9,999
05 ❑ 10,000 - 49,999
06 ❑ More than 49,999

Publisher reserves the right to determine qualification for FREE subscriptions.

▼ FOLD HERE ▼

▼ FOLD HERE ▼

PLEASE MAKE SURE YOU'VE:
• Signed and dated the form
• Filled out the form completely
• Fax to: 1-650-321-3818 or fold, tape and mail (*please do not staple*)

BUSINESS REPLY MAIL
FIRST-CLASS MAIL PERMIT NO 317 PALO ALTO CA

POSTAGE WILL BE PAID BY ADDRESSEE

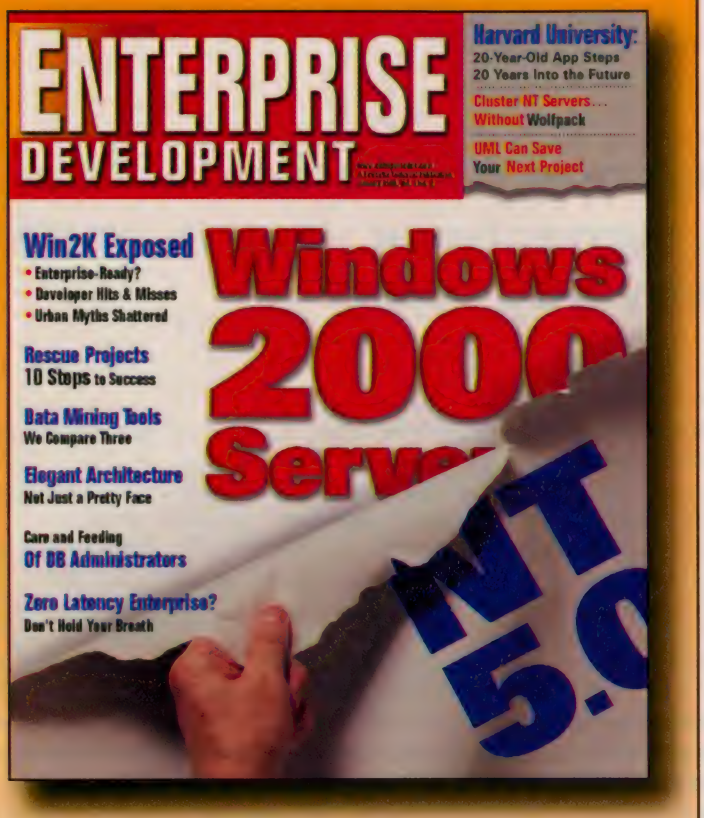
ENTERPRISE DEVELOPMENT

Attn: New Subscriptions
209 HAMILTON AVENUE
PALO ALTO CA 94301

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

FREE Subscription
To Qualified Applicants

Enterprise Development magazine



In every issue:

- **Case studies that don't leave out the "gotchas"**
- **How to get more from programming teams....and your middleware**
- **How to integrate software with ERP systems such as SAP and Peoplesoft**
- **Achieving and documenting a ROI for improvements in the development process**

All the information you need to architect enterprise apps and better manage the development process

Until now, no single magazine has met the needs of enterprise development managers. They've had to scour countless publications aimed at CIOs or coders in search of solutions to their specific problems.

Finally there's a magazine that cuts through all the confusion to bring the right answers together under one cover: *Enterprise Development.* **And it's free to qualified subscribers.***

Enterprise Development is written by and for IS professionals responsible for the design and management of application development projects. Its an independent source that will investigate and evaluate what to use in enterprise development projects. It examines the good, bad and ugly of new tools, technologies and techniques, and includes case studies on everything from n-tier MTS applications to simple extensions of legacy applications out through the Web. *Enterprise Development* is designed to help projects scale-up in terms of users supported, without also scaling-up the budget and time-to-market.

Don't waste another minute–reserve your FREE SUBSCRIPTION today. Here's how:

- **Complete the subscription form on the facing page, fold, seal, stamp, and mail**
- **Or fax your completed form to 1-650-321-3818**
- **Or subscribe online at www.enterprisedev.com**

*Publisher reserves the right to determine qualification for free subscriptions.

FTP FAWCETTE TECHNICAL PUBLICATIONS

209 Hamilton Avenue
Palo Alto, CA 94301-2500
Phone: 650.833-7100

Windows NT is a registered trademark of Microsoft Corporation.
PeopleSoft is a registered trademark of PeopleSoft Inc.
All other brand names or trademarks are the property of their respective companies.

Get Semaphores in Line

Java's mixed blessings strike again. It's easy to create a semaphore, but it might not behave the way you want

by Rick Grehan

When discussing the virtues of Forth, a friend of mine once said, "The good news is that you can do anything you want. The bad news is that you have to do it all yourself." The same is true of semaphores in Java. As I stated last month, Java's concurrency mechanism, based on monitors, is all you need to implement semaphores, mutexes, critical section objects, and so on. That's the good news. The bad news is that if we want "proper" behavior from whatever concurrency mechanism we build, we have to do it ourselves.

Last month, as part of my discussion of priority inversion, I showed a simple binary semaphore that worked like a lock. It was simple not only in the sense that its code was brief, but also because it permitted only one thread at a time to own the lock. This aspect makes it suitable for situations in which a resource (such as a file or a printer) can only be accessed by a single thread at a time.

However, some resources permit more elaborate forms of access control. For example, a database server might support several simultaneous client connections. Because a binary semaphore allows only one thread (or task) through at a time, it's insufficient to handle resources that allow multiple simultaneous accesses. Fortunately, a slight modification to the binary semaphore gets us what we need—the counting semaphore.

Can You Count?

The modification starts with a change in the data type of the variable that the binary semaphore used to maintain its internal state. If you remember from last month, that internal variable was a boolean, `isLocked`. If `isLocked` was true, the semaphore was locked—any thread trying to acquire (lock) the semaphore had to wait. If `isLocked` was false, a thread attempting to acquire the semaphore succeeded, setting `isLocked` to true. When a thread unlocked the semaphore, it set `isLocked` to false and notified any threads waiting on the semaphore.

Turning the binary semaphore into a counting semaphore requires changing `isLocked` into a counter (an integer), and slightly modifying the semaphore's internal behavior. The resulting class is shown Listing 1.

The counting semaphore implements two methods—`semaWait()` and `semaSignal()`—corresponding to the `lock()` and `unlock()` calls of the binary semaphore. The formal definition of a semaphore uses the `wait()` and `signal()` methods. However, because the Object class defines a final method called `wait()`, I've made the semaphore's method `semaWait()`. To keep things consistent, the semaphore's signal method is `semaSignal()`.

A thread calls `semaWait()` to gain access to whatever resource the semaphore is guarding. When a thread calls `semaWait()`, that method—which is synchronized—checks the value of the semaphore's count variable. If count is greater than zero, count is decremented and the calling thread proceeds. Otherwise, the thread waits.

A thread calls signal to indicate that it has "released" the semaphore. This call increments the semaphore variable and issues a `notify()` to release a waiting thread. If no threads are waiting, the `notify()` has no effect.

A counting semaphore is like a ticket booth with a fixed number of tickets available. A thread that wants to enter the resource guarded by the booth must take a ticket. If one is available, the ticket count is decreased by one, and the thread proceeds. If no tickets are available, waiting threads queue up at the booth. Whenever a thread exits, it turns in its ticket at the ticket booth. The available ticket can be used to admit one of the waiting threads. What's important is that the semaphore starts out with an initial count that indicates the number of `semaWait()` calls it can sustain before a thread is

Rick Grehan is a programmer/writer in the Discover Products group at Metrowerks Inc. Rick was a Senior Editor at *BYTE* magazine (where he wrote the JavaTalk column) and at *Computer Design* magazine. He can be reached at rgrehan@top.monad.net.

Listing ① More Than One at a Time The binary semaphore from last month's column has been modified to be a counting semaphore. This enables more elaborate forms of access control to resouces such as a database server. With the counting semaphore multiple simultaneous accesses are possible.

```
//
// Class to implement a counting Semaphore.

public class Semaphore
{
//  Create a semaphore with an initial count.
//  This determines the number of threads
//  that will be allowed to "pass through"
//  the semaphore before it closes the door.
  public Semaphore(int initcount)
  {
    count=initcount;
  }

// semaWait
// Attempt to acquire (lock) the semaphore.
  public synchronized void semaWait()
  {
    while(count<=0)
    { try {
        this.wait();
      }
      catch (InterruptedException ex) { }
    }
    count--;
  }

// semaSignal
// Release (unlock) the semaphore. Notify
// a waiting thread on the way out.
  public synchronized void semaSignal()
  {
    count++;
    this.notify();
  }

  private int count;
}
```

suspended. `semaWait()` decrements that count, `semaSignal()` increments it.

What's the Catch?

Although the counting semaphore we've defined works, there's a *potential* catch. I say potential because—depending on the application—it might not matter. The catch is that we really have no control over which thread is released when `semaSignal()` is called. In most cases, semaphores are implemented entirely by the operating system, which has a complete knowledge of which threads (or tasks) might be waiting on the semaphore. The operating system is therefore in a position to make an "intelligent" decision about which thread gets released when the semaphore is signaled. Because we're implementing a semaphore "on top of" the concurrency mechanism in Java, the VM scheduler has no idea that what we've built is a semaphore. We have no control over which thread it decides to wake up in response to a `notify()`; we have to create that control ourselves. (This is the bad news part I mentioned at the start.)

We can presume that the highest-priority thread waiting on the semaphore will be the next to run. However, this might not be the case if several threads are of the same priority and are waiting on the semaphore. The scheduler's choice of the next thread to release might be entirely random (from our point of view).

Some resources permit more elaborate forms of access control

The result could be something akin to "starvation." For example, a thread could arrive at the `semaWait()` call and be forced to wait. Another thread and then another could arrive at the same `semaWait()` call and also be forced to wait. When a `semaSignal()` is finally called, it's possible that the thread that arrived at `semaWait()` first will not be the first thread allowed to proceed. Because we don't know how the scheduler will release threads, we don't know how long a thread will have to wait. This is hardly fair—we need a queued semaphore

Queued Semaphore

Simply put, a queued semaphore has an associated queue. At first glance, associating a queue with a the semaphore object appears to be simple. If a thread arrives at the `semaWait()` and the semaphore is "owned" (has no available connections), the thread simply goes into a queue. Whichever thread calls `semaSignal()` wakes up the thread at the head of the queue—removing that first queued element.

On second glance, however, we realize we need to be a little more clever. There are really two things that must be governed by synchronized keywords: The semaphore itself and the queue. It's possible for one thread to be attempting to call `semaWait()` while another is calling `semaSignal()`. In that case, the first thread will be putting an element on the queue while the second thread will be taking an element off.

Listing ② Threads in a Row The queued semaphore class acts like a ticket booth. A limited number of threads are allowed in depending on available tickets. When a thread exits, it returns its ticket, and the thread at the head of the line is allow in.

```
//
// Class to implement a queued Semaphore.
// Threads that enter the semaphore are
// queued, so that they are permitted
// access in the order they requested it.

public class queuedSemaphore
{

//
// Create a queued semaphore with an
// initial count.
public queuedSemaphore(int initcount)
{
  count=initcount; // Initial count
  lQueue=new Queue();  // Make a new Queue
}

//
// semaWait
// Obtain a lock. If we don't get one,
// we have to go sit on the queue.
public void semaWait()
{
  Semaphore mySema;  // Semaphore object
```

```
  synchronized(this)
  {
    // See if access is allowed
    if(count<=0)
    {
      // It's locked. Create a new
      // counting semaphore and
      // put it on the queue.
      mySema=new Semaphore(0);
      lQueue.put(mySema);
    }
    else
      // It's not locked. Go ahead and
      // lock it and return without
      // going on queue
    { count--;
      return;
    }
  }
  // Now step outside of synchronized block.
  // We do this because the wait we are about
  // to issue is on the Semaphore object,
  // NOT the queued semaphore object.
  mySema.semaWait();
}

// semaSignal
// Signal the queued semaphore
public void semaSignal()
{
  Semaphore mySema;

  synchronized(this)
  {
    // See if anyone is on the queue.
    // If so, remove the queue element and
    // release whoever is waiting.
    // Notice that we don't *really*
    // signal...we pass the un-signalled
    // condition on to the next thread.
    // In that way, the count variable is
    // always modifed from within a
    // synchronizated block.
    if(lQueue.isEmpty()==false)
    { mySema=(Semaphore)lQueue.get();
      mySema.semaSignal();
    }
    else
      // No one one queue. Unlock
      count--;
  }
}

  private int count; // Semaphore's count
  private Queue lQueue; // Queue to govern lock
}
```

Figure 1 Queued Elevators In this run of a queued semaphore in action, the initial count of the semaphore is set to one, so that only one elevator at a time is allowed to move. Each elevator "takes its turn" moving, and the left-to-right sequence is preserved throughout the execution of the application.

We also need some way to suspend and resume threads without resorting to calls to `suspend()` and the associated `resume()`. There are two related reasons for this. Although I still use JDK 1.1.x, `suspend()` is deprecated in JDK 1.2. The other reason (which explains the deprecation) is that calling `suspend()` from within a synchronized object will not release the monitor on that object. A thread `suspend()`ed in a synchronized method will lock that object until an outside thread `resume()`s the suspended thread to exit the synchronized method. On the other hand, `wait()` releases the monitor, which is why calling `wait()` from within a synchronized method is safe. Fortunately, we already have a class that takes care of all of this: the semaphore class we just defined.

Under the Hood

The code for the `queuedSemaphore` class is shown in Listing 2. When a thread calls `semaWait()`, it immediately falls into a synchronized block. Within that block, the thread tests the state of the semaphore's associated count variable. If it's greater than zero, the thread simply decrements it and exits the method, leaving the synchronized block and releasing the associated monitor. In this case, the queued semaphore is working like the counting semaphore.

If the count is less than or equal to zero, the thread creates a new counting semaphore object and places that object in the associated queue. Next, the thread leaves the synchronized block and calls the `semaWait()` method. Don't get confused. There are two `semaWait()` methods here because we're working with two semaphores. The `semaWait()` being called at this point is associated with the counting semaphore that was just placed on the queue.

It is important to remember that the thread must call the counting semaphore's `semaWait()` from outside the synchronized block. The `semaWait()` is also synchronized and includes a `wait()` call. That `wait()` call will release the monitor on the counting semaphore object, but *not* the queued semaphore object. If the call to the counting semaphore's `semaWait()` is inside the queued semaphore object's synchronized block, a single thread will block any other thread's ability to complete a call into the queued semaphore's methods.

A thread entering the queued semaphore's `semaSignal()` will first test to see if anything is on the queue. If so, the thread removes the head counting semaphore, signals it, and then exits. Notice that, in this case, the signaling thread does not decrement the signaling semaphore's count. In a sense, it "passes" the ticket to the thread that has just been released from the queue. If, however, the queue is empty, the thread calling `semaSignal()` simply increments the count variable and exits.

In Action

I've modified the elevator program from last month to demonstrate the queued semaphore in action. The Listing 3 also includes the queue

Code Online

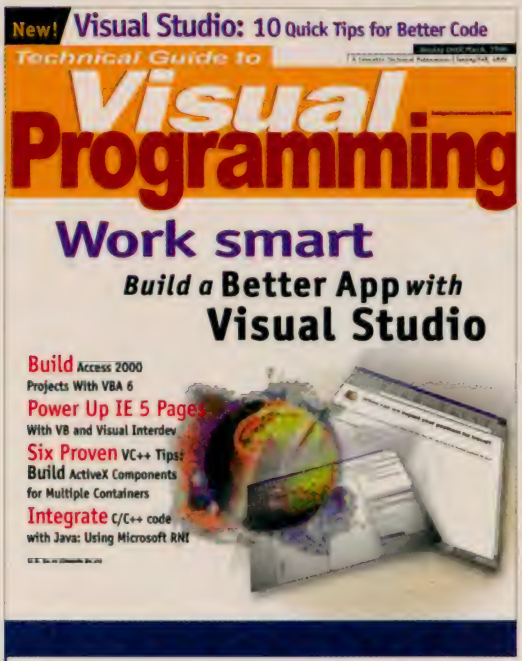
All the code in this article, including the elevator program in Listing 3, is available for free at www.java-pro.com. Use the Locator+ codes shown below to download the code. Free online registration is required. See the Online Resources box in "Letters to the Editor" for more information.

Get Semaphores in Line
Registered Locator+: **JP0399**
Premier Locator+: **JP0399IT**

NEW

Introducing the newest quarterly publication from Fawcette Technical Publications

Technical Guide to Visual Programming


New! Visual Studio: 10 Quick Tips for Better Code
Technical Guide to
Visual Programming
Work smart
Build a Better App with Visual Studio
Build Access 2000 Projects With VBA 6
Power Up IE 5 Pages With VB and Visual Interdev
Six Proven VC++ Tips: Build ActiveX Components for Multiple Containers
Integrate C/C++ code with Java: Using Microsoft RNI

WORK SMARTER. BUILD FASTER APPS. SAVE TIME. SAVE MONEY.

Here's your chance to get all the know-how you need to build production-quality applications with Visual Studio 6 and other visual programming tools, all in one magazine. Published quarterly for professional developers and programming teams, the *Technical Guide to Visual Programming* will give you key information designed to help you work faster, better and smarter.

You also get objective product reviews and tons of technical tips in each issue.

Full money-back guarantee: If at any time you find *Technical Guide to Visual Programming* doesn't increase your productivity, doesn't tip you off to the newest programming tools, or doesn't publish helpful utilities you can actually use — write us and we'll refund your full subscription price. No questions asked.

Here's a sampling of what you'll find in our premiere issue:

- How to leverage VS in your apps
- Supercharge VB's forms with ActiveX
- Creating ATL controls quickly and easily
- Creating controls for multiple containers

ORDER YOUR COPY TODAY

$24.95 Four issues annually

To start your subscription, simply return the attached card, or for faster service call:

1-800-848-5523

Outside the U.S., call 650-833-7100

FTP FAWCETTE TECHNICAL PUBLICATIONS
209 Hamilton Ave. Palo Alto, CA 94301
www.devx.com

Please allow 4-6 weeks for delivery of first issue. International subscriptions must be payable in U.S. dollars plus postage; Canada/Mexico add $6/year for surface delivery; all other countries add $15/year for airmail delivery. Visual Studio and Visual Basic are registered trademarks of Microsoft Corporation. All other trademarks are property of their respective owners.

class used by the queued semaphore class. This time, I've set up six elevators, started in sequence with some delay between the start times so that the behavior of the queued semaphore is more apparent. Each elevator enters the queued semaphore, does some moving and some sleeping, and then signals the semaphore (see Figure 1).

Access to the queued semaphore (which is what permits an elevator to move) is handed in bucket-brigade fashion down the line of elevators. You can see a wave motion as each elevator, in turn, is given access to the semaphore. "In turn" is the key. The elevators are queued and released in sequence. You can experiment with letting more than one elevator through the semaphore at a time by modifying the initial count given to the queued semaphore's constructor.

Lining Up

As effective as the queued semaphore is, there's still room for modifications. For example, the queue elements don't need to be counting semaphores, they can just as easily be binary semaphores. It would also be helpful to provide the ability to make the queue a priority queue. This way, high priority threads would move to the front, but sequential order among threads of equal priority would be preserved. Finally, you can add information to queue elements that timestamp when an item goes on the queue and allow a thread to specify how long it's willing to wait. A daemon thread can prowl the queue, kicking off threads whose wait time has expired. ■

Listing 3 Taking Their Turn To demonstrate queued semaphores, this code starts up six elevators in sequence with some delay between the start times so that the behavior of the queued semaphore is more apparent. You can see a wave motion as each elevator is given access to the semaphore.

```
import java.awt.*;
import java.lang.Thread;

//
// Class to implement Elevator application for
// illustrating queued semaphores.
// Rick Grehan
// JavaPro Magazine
// March. 99
//
public class ElevatorApp extends Frame
{
  public ElevatorApp()
  { setTitle("Queued Semaphore Demo");
    c=new Canvas();
    add("Center",c);
  }

  public void StartElevator()
  {
    // Create a new Queued Semaphore Object
    // The number given to the constructor sets
    // the number of elevators allowed
    // "through" at any given time.
    // Change this number to 2 or 3
    // and watch the results.
    qs=new queuedSemaphore(1);

    // Create each elevator...waiting a little
    // after each creation.
    // This will cause the elevators to "queue
    // up" for the lock. (Note: The wait
    // values I've set provide the proper
    // timings on my system to see the effect
    // of the queued semaphore. You may
    // have to tweak them for your system.)
    e1 = new Elevator(c,
      10, Thread.NORM_PRIORITY,
      Color.black, qs);
    e1.start();
    sleepSome();

    e2 = new Elevator(c,
      30, Thread.NORM_PRIORITY,
      Color.red, qs);
    e2.start();
    sleepSome();

    e3 = new Elevator(c,
      50,Thread.NORM_PRIORITY,
      Color.blue, qs);
    e3.start();
    sleepSome();

    e4 = new Elevator(c,
70,Thread.NORM_PRIORITY,
      Color.green, qs);
    e4.start();
    sleepSome();

    e5 = new Elevator(c,
      90,Thread.NORM_PRIORITY,
      Color.orange, qs);
    e5.start();
    sleepSome();

    e6 = new Elevator(c,
      110,Thread.NORM_PRIORITY,
      Color.yellow, qs);
    e6.start();
  }

  public void sleepSome()
  {
    try { Thread.sleep(80); }
    catch (InterruptedException ex)
    { }
  }

  public static void main(String[] args)
  {
    ElevatorApp f = new ElevatorApp();
    f.setSize(300,200);
    f.show();

    // Wait for screen to get drawn
    try {
      Thread.sleep(2000);
    }
    catch (InterruptedException ex)
    { }

    f.StartElevator();
    // Wait for a while, then end.
    try {
      Thread.sleep(50000);
    }
    catch (InterruptedException ex)
    { }
    System.exit(0);
  }

  private Canvas c;
  private Elevator e1, e2, e3, e4, e5, e6;
  private queuedSemaphore qs;
}



//
// Elevator class.
// Rick Grehan
// JavaPro Magazine
// Dec 1998
//
import java.awt.*;

class Elevator extends Thread
```

```
{
  // Constructor for Elevator.
  // The argument "place" selects which
  // "position" the elevator is created
  // in (horizontal position, that is).

  public Elevator(Canvas c, int place,
    int priority, Color col,
    queuedSemaphore qsema)
  {
    box=c;     // Get reference to Canvas
    x=place;   // Set location onscreen
    p=priority;  // Set thread's priority
    clr=col;   // Set box color
    qs=qsema;    // Set queued semaphore
  }

  public void run()
  {
    double k=.5;

    // Set our priority
    try {
      setPriority(p);
    }
    catch (IllegalArgumentException ex)
    { }

    // Draw the first instance of the elevator.
    Graphics g=box.getGraphics();
    g.setColor(clr);
    g.fillRect(x,y,XSIZE, YSIZE);
    g.dispose();

    // We won't run forever. Just for a
    // few thousand iterations.
    for(int i=1; i<=3000; i++)
    {
      // Each elevator locks the queued
      // semaphore, then does some moving
      // and sleeping. It then signals
      // (unlocks) the queued semaphore,
      // and sleeps a little more (so that
      // other threads can get at the
      // semaphore).
      qs.semaWait();

      for(int j=1; j<10; j++)
      {
        move();

        try { Thread.sleep(60); }
        catch (InterruptedException ex) { }
      }

      qs.semaSignal();

      try { Thread.sleep(5); }
      catch (InterruptedException ex) { }
    }
  }

//
// move
// Move the elevator one pixel up or down.
  public void move()
  {
    Graphics g=box.getGraphics();
    Dimension d=box.getSize();

    g.setColor(clr);

    // Erase the old
    g.clearRect(x,y,XSIZE,YSIZE);
    y+=yincr;  // Advance elevator

    // If the elevator has crashed into
    // the top, set its location to the
    // top and reverse direction.
    if (y<0)
      { y=0;
        yincr=-yincr;
      }

    // If the elevator has crashed into
    // the bottom, set it's location to
    // the bottom and reverse it's direction.
    if (y+YSIZE>=d.height)
    { y=d.height-YSIZE;
      yincr=-yincr;
    }

    // Draw the new rectangle and exit
    g.fillRect(x,y,XSIZE,YSIZE);
    g.dispose(); // Release Graphics object
  }

  private Canvas box;
  private static final int XSIZE=10;
  private static final int YSIZE=10;
  private int x=0;
  private int y=0;
  private int yincr=2;   // Move increment
  private int p;     // Thread priority
  private Color clr;   // Box color
  private queuedSemaphore qs;
}



//
// Queue using Vector class
// Based on code found in "Industrial
// Strength Java" by Luke Cassady-Dorian
// et. al, (New Riders Publishing)
import java.util.*;

public class Queue extends java.util.Vector
{

//
// get
// Returns the head object from the queue
// and, in so doing, removes the object
// from the queue. Returns the object
// reference
public Object get()
{
  Object temp=super.firstElement();
  super.removeElementAt(0);// Remove queue
                           // node
  return(temp);              // Return object
                           // ref.
}

//
// put
// Attaches an object to the "end"
// of the queue
public void put(Object payload)
{
  super.addElement(payload);
}

//
// peek
// Returns reference to head object without
// removing the corresponding node from
// the queue
public Object peek()
{
  return super.firstElement();
}

//
// In "Industrial Strength Java", the authors
// redifined the isEmpty() method to be
// empty(). We'll just inherit isEmpty()
}
```

Portal Kombat

Take on the dark sorcerers of Yahoo! with a servlet-driven portal site using JavaServer Pages

by Larry O'Brien

Portals, sites combining a powerful search capability with a predefined hierarchy of categories, have emerged as natural home pages for the majority of Web users. This is in contrast with the original concept of personal "home" pages, but the Web has grown so fast that defining, organizing, and reorganizing a categorization is no longer a useful personal project. Portal sites like Yahoo! and Excite provide an entrance into the whole Web, and serve millions of people with diverse needs. I think '99 will be the year of niche portals.

A niche portal is not based on lowest-common denominator categories, but rather on categories that define some work-dominating topic. A niche portal starts at the second or third tier of Yahoo!'s hierarchy. The Java niche portal I developed for this month's column is shown in Figure 1. The structure is straightforward: News on the left, the latest posts from Sun and other vendors on the right, and categories of interest to Java programmers in the center. Add a search capability and a couple of ads in the top center and voilá—a niche portal. The Java niche portal we'll develop is both a resource and an example of driving a Web site with advanced technologies such as database-stored content and JavaServer Pages (JSP). Such niche portal sites might attract tens of thousands of visitors—making the difference between a domain that languishes and one that achieves commercial success. A well-targeted portal might even compete with Yahoo! for the loyalties of desirable audiences (such as Java programmers).

As discussed in last month's column, the key to creating a dynamic site is to move content into a database. It's difficult enough to update content, updating content within

Figure 1 Finding Your Niche Creating a niche portal can help users find the information really useful to them without drilling down through several layers as they would in a Yahoo! search.

Larry is an independent trainer and consultant and developed *Bruce Eckel's Hands-On Java CD*. He is the chair of the "Internet and Intranet Development" track for the Software Development conference. Reach him at larryo@altnetinc.com.

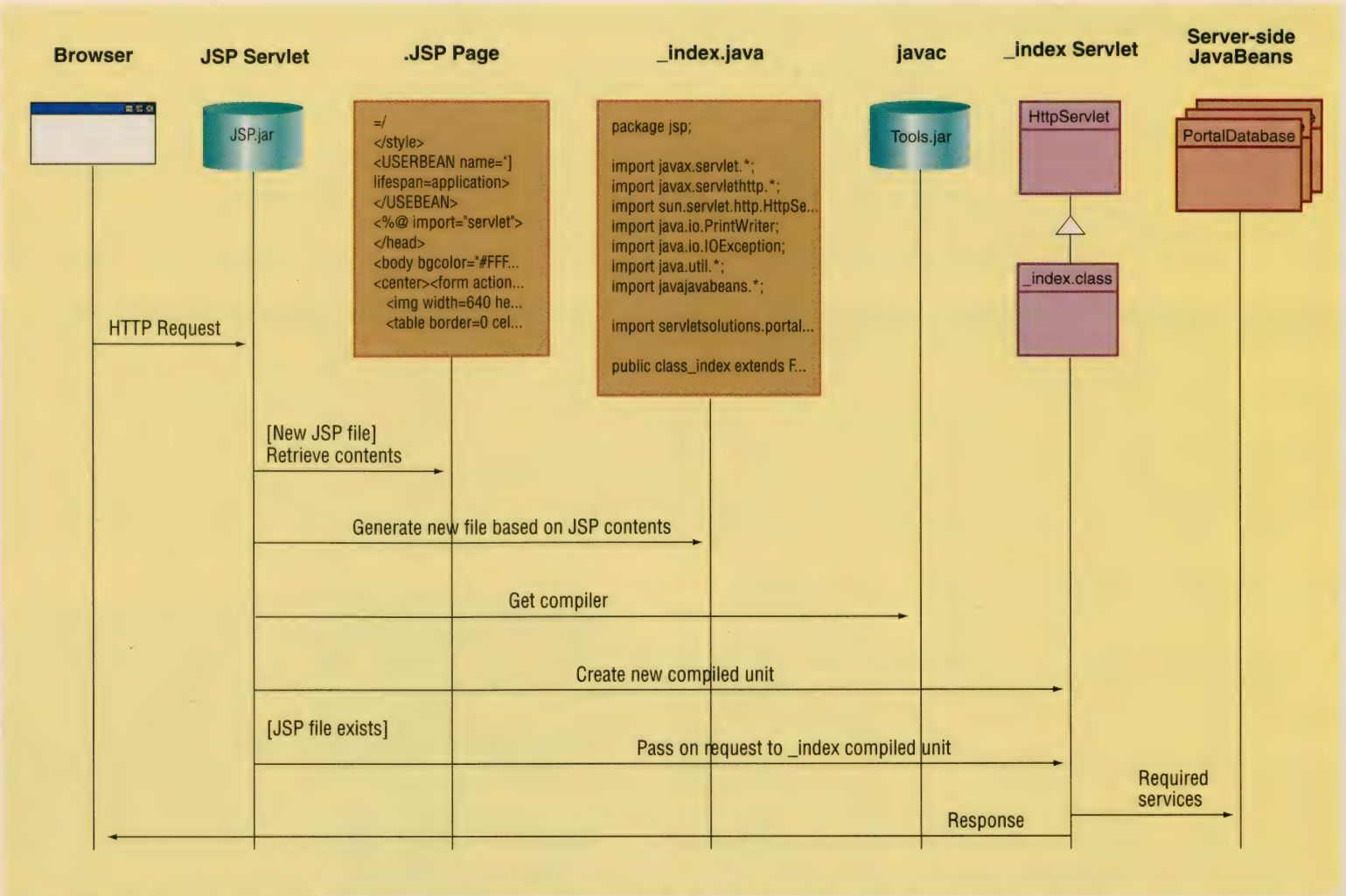


Figure ② JSP Sequence Diagram JSP pages are stored with an extension of .jsp and the servlet server is configured so a special page-compilation servlet will be invoked whenever such a page is accessed. The first time the page compilation servlet reads a .jsp file, it generates a servlet that combines the HTML tags in the .jsp file with the Java code embedded within the <% %> brackets. It then compiles the servlet and chains the request to the newly created servlet.

a delicate framework of tables or DIV SPANs is simply too much work. In previous columns, we've created dynamic content directly from servlets. The pages created have been extremely plain, and the servlet code has still been filled with HTML code written to the ServletResponse's PrintWriter. JSP, a technology from Sun that should be publicly available by the time you read this, allow us to avoid writing such code while providing a well-architected component model for server-side processing. This column is based on the 0.92 JavaServer Pages specification, there might have been minor changes in JSP syntax since the time of writing.

Freedom with JSP

JSP allows you to use Java as a scripting language within an HTML file by enclosing code within paired <% %> brackets. Additional server-side functionality can be accessed by adding the `USEBEAN` tag to declare that a particular JavaBean is used. JSP pages are stored with an extension of .jsp, and the servlet server is configured so a special page-compilation servlet (provided by the servlet server) will be invoked whenever such a page is accessed (see Figure 2). The first time the page compilation servlet reads a .jsp file, it generates a servlet that combines the HTML tags in the .jsp file with the Java code embedded within the <% %> brackets. It then compiles the servlet and chains the request to the newly created servlet. Future requests go directly to the generated servlet. In effect, it automatically creates the HTML portion of a servlet, leaving the behavior up to either the scripting code within the page or, better yet, external JavaBeans.

Why "better yet"? Because separating presentation (in this case, the HTML code in the .jsp file) from application logic (JavaBeans) is one of the fundamental principles of software engineering. Although HTML design tools do not currently support JSP technology, they are likely to once JSP 1.0 is released by Sun. Because JavaBeans can expose information about themselves, and even have a visual representation at design time, an HTML design environment could allow page designers to drag-and-drop JavaBeans directly onto a page, precisely positioning the Bean for the best effect.

The Java niche portal structure is straightforward

Creating our Portal

I mocked up the "Java Del Porto" niche portal in Macromedia Dreamweaver. Because Dreamweaver does

Listing 1 Creating Dynamic Content This standard Java code, wrapped in println() statements, creates the dynamic content on the right and left panes of the Java Del Porto page.

```
<%
  Iterator i = portal.getLatestHeadlines("News", 3);
  while(i.hasNext()){
    Headline it = (Headline) i.next();
    out.println("<tr>");
    out.println("<td valign=top><b>&#183;</b></td>");
    out.println("<td valign=\"top\"><p><small>");
    out.println("<a href=\"");
    out.println(it.getTargetURL());
    out.println("\">");
    out.println(it.getHead());
    out.println("</a><br>");
    out.println(it.getAbstract());
    out.println("</small></p></td></tr>");
  }
%>
```

Listing 2 Classy Portals The Portal and PortalDatabase classes retrieve the most recent headlines in the specified category from the database. The PortalDatabase class is a modified version of the WhatsNewDatabase class presented in last month's column.

```
package servletsolutions.portal;
import java.util.*;

public class Portal{
  PortalDatabase db = null;

  public Portal(){
    try{
      db = PortalDatabase.getDatabase();
    }catch(Exception e){
      System.err.println(e);
    }
  }

  public synchronized Iterator getCategories(){
    Iterator c = db.getCategories();
    return c;
  }

  public synchronized Category getCategory(int i){
    return db.getCategory(i);
  }

  public synchronized Iterator getLatestHeadlines(String
    category, int i){
    return db.getLatestHeadlines(category, i);
  }

}

PortalDatabase.java
package servletsolutions.portal;

import java.sql.*;
import java.util.*;

class PortalDatabase{
    static final String driverName =
      "sun.jdbc.odbc.JdbcOdbcDriver";
    static final String dbName =
      "jdbc:odbc:news";
    static final String user = "";
    static final String pass = "";

    static PortalDatabase singleton;
    //Can't use static initializer for singleton because
    //constructor throws exceptions

    private PortalDatabase() throws
    ClassNotFoundException{
      Class.forName(driverName);
    }

    static PortalDatabase getDatabase() throws
      ClassNotFoundException{
      if(singleton == null){
        synchronized(PortalDatabase.class){
          if(singleton == null){
            singleton = new PortalDatabase();
          }
        }
      }
      return singleton;
    }

  protected Iterator getCategories(){
    Iterator ret = null;
    try{
      Vector cats = new Vector();
      Connection cnctn =
        DriverManager.getConnection(dbName);
      Statement s = cnctn.createStatement();
      String query = "SELECT * FROM CATEGORIES";
      ResultSet r = s.executeQuery(query);
      while(r.next()){
        String name = r.getString("Name");
        String desc = r.getString("Description");
        long id = r.getLong("ID");
        long   parent = r.getLong("ParentCategory");
        Category c = new Category(id, name, desc, parent);
        cats.add(c);
      }
      ret = cats.iterator();
    }catch(SQLException se){
      System.err.println("Error retrieving categories.");
      System.err.println(se);
    }catch(RuntimeException r){
      System.err.println("RE " + r);
      throw r;
    }
    return ret;
  }

  protected Category getCategory(int i){
    Category ret = null;
    try{
      Connection cnctn =
        DriverManager.getConnection(dbName);
      Statement s = cnctn.createStatement();
      String query = "SELECT * FROM CATEGORIES WHERE ID = "
         + i + ";";
      ResultSet r = s.executeQuery(query);
      if(r.next()){
        String name = r.getString("Name");
        String desc = r.getString("Description");
        long id = r.getLong("ID");
        long   parent = r.getLong("ParentCategory");
        ret = new Category(id, name, desc, parent);
      }
    }catch(SQLException se){
      System.err.println("Error retrieving categories.");
      System.err.println(se);
    }catch(RuntimeException r){
      System.err.println("RE " + r);
      throw r;
    }
    return ret;
  }

  protected Category getCategory(String name){
    Category c = null;
```

Continued on next page.

Continued from previous page.

```
  try{
    Connection cnctn =
      DriverManager.getConnection(dbName);
    Statement s = cnctn.createStatement();
    String query = "SELECT * FROM CATEGORIES WHERE Name = "
      + name + ";";
    ResultSet r = s.executeQuery(query);
    if(r != null){
      r.next();
      String desc = r.getString("Description");
      int  id = r.getInt("ID");
      int    parent = r.getInt("Parent");
      c = new Category(id, name, desc, parent);

    }
  }catch(SQLException se){
    System.err.println(se);
  }
  return c;
}

protected Iterator getLatestHeadlines(String category,
  int i){
    Vector items = new Vector();
    try{
      Connection cnctn =
        DriverManager.getConnection(dbName);
      Statement s = cnctn.createStatement();
      StringBuffer query = new StringBuffer("SELECT * FROM
        Headlines, Categories ");
      query.append("WHERE (((Categories.Name)='" +
        category + "') ");
      query.append("AND (Headlines.Category =
        Categories.ID)) ORDER BY Headlines.Posted DESC;");
      ResultSet r = s.executeQuery(query.toString());
      for(int cnt = 0; cnt < i && r.next(); cnt++){
        String head = r.getString("Head");
        String abs = r.getString("Abstract");
        String targ = r.getString("TargetURL");
        Headline h = new Headline(head, abs, targ);
        items.add(h);
      }
    }catch(SQLException se){
      System.err.println(se);
    }
    return items.iterator();
  }
}
```

TechTIP

Right-Clicks on a Single Button Mouse

Those of us with a two- or three-button mouse are familiar with the idea of a right-click, but how can you detect which button is pressed from your Java code? And how do you cater for similar functionality with a one-button mouse and keep your applet or application platform independent? Fortunately, the `InputEvent` modifier flags serve double duty. Clicking the right button or holding down the Alt key while clicking the left button have the same effect. By masking the event modifiers in a `mouseClicked` method, you can permit the user to press Shift, Ctrl, Meta or Alt keys:

```
import java.awt.*;
import java.awt.event.*;
public class YourClass implements MouseListener {
  // Your constructors and methods here
  public void mouseClicked(MouseEvent ev) {
    int mask = InputEvent.BUTTON1_MASK - 1;
    int mods = ev.getModifiers() & mask;
    if (mods == 0) {
      // Left button clicked
    }
    else {
      // Right button clicked
    }
  }
}
```

Although this code does not distinguish between the right and center buttons on a three-button mouse, it could be extended to do so.

–Tom Yarker

not understand JSP's syntax, this code for the Head block of the home page was modified in a text editor:

```
<head>
<title>Java Del Porto</title>
<USEBEAN name="portal"
type=servletsolutions.portal.Portal
lifespan=application>
</USEBEAN>
<%@ import="servletsolutions.portal.*" %>
<link rel="stylesheet" href="portal.css">
</head>
```

In addition to standard HTML tags for the title and a link to a cascading stylesheet, there are two special JSP tags. The first is the `USEBEAN` tag, which makes a Bean available for use later in the page.

Listing 3 Heads Up The Headline class is a simple encapsulation of the data in the Headline database table. It defines a headline as consisting of a head, an abstract, and a target URL.

```
package servletsolutions.portal;

public class Headline{
  String head;
  String abs;
  String targetURL;

  Headline(String head, String abs,
String targetURL){
    this.head = head;
    this.abs = abs;
    this.targetURL = targetURL;
  }

  public String getHead(){
    if(head == null)
      head = "";
    return head;
  }
  public String getAbstract(){
    if(abs == null)
      abs = "";
    return abs;
  }
  public String getTargetURL(){
    if(targetURL == null)
      targetURL = "";
    return targetURL;
  }
}
```

Resources

JSP Servers

JRun Scripting Toolkit; Live Software:
www.livesoftware.com

ServletExec; New Atlanta Communications:
www.newatlanta.com

WebSphere; IBM Corp:
www.ibm.com/websphere

More JSP Info

http://java.sun.com/products/jsp/jsp092.html

In this case, a `servletsolutions.portal.Portal` Bean is instantiated and assigned to the variable `portal`. This Bean is declared as valid for the application's entire lifespan. Other options for this parameter are `Page` and `Session`. The second JSP tag shown is an import statement using the special tag "`<%@ `" (note that a space is required after the `@`).

The embedded Java code that creates the dynamic content on the left and right sides of the page is shown in Listing 1. As you can see, aside from the enclosing `<% %>` brackets, it's standard Java that loops over an iterator, outputting HTML code (specifically, a table row). This was the only time I had to wrap HTML code in println statements and, strictly speaking, the println statements weren't necessary. I could have interspersed Java code in `<% %>` brackets and HTML outside. However, the code's behavior is clearer when it is in a single block and wrapping a few lines of HTML is not an onerous task.

The code in Listing 1 refers to two other classes in the `servletsolutions.portal` package. The `Portal` class is a thin wrapper of the `PortalDatabase` class, both classes are shown in Listing 2. If the `PortalDatabase` class looks familiar—it should! It's the `WhatsNewDatabase` class presented last month (minus some methods specific to last month's servlet). The `PortalDatabase.getLatestHeadlines(String category, int i)` method is called by JSP, and retrieves the *n* most recent Headlines in the specified category from the database. The `Headline` class, shown in Listing 3, is a simple encapsulation of the data in the Headline database table, which defines a headline as consisting of a head, an abstract, and a target URL. Back in Listing 1, you can see the `Headline` class accessed by the Java code embedded in the JSP file; the headline, link, and descriptive text of the last several headlines in a particular category are transformed into HTML code and embedded in a table.

The Future of JSP

It's important to realize that all of this behavior takes place on the server-side. What is eventually passed back to the client is nothing but HTML, which of course can be as simple or as complex as desired. Naturally, this increases server-side loading, and all the normal concerns about servlets and database scaling must be taken into account.

The success of JSP seems assured. Microsoft's Active Server Pages, which obviously inspired the JSP effort, drive some of the most popular sites on the Web, and JSP is far more portable. It is only a matter of time before JSP becomes available for all servers with servlet capability. Several third-party vendors are tracking the JSP specification and making JSP servers available (see "Resources"). The architecture of JSP is also flexible enough to support new languages or component technologies. Just as one of the more interesting things about the servlet architecture is that it is not locked into the Hypertext Transport Protocol, one of the more interesting things about JSP technology is that it is not locked into Java as the scripting language, JavaBeans as the component model, or a servlet as the end result of page compilation. ■

Code Online

All the code in this article is available for free at www.java-pro.com. Use the Locator+ codes shown below to download the code. Free online registration is required. See the Online Resources box in "Letters to the Editor" for more information.

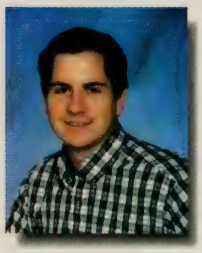
Portal Kombat
Registered Locator+: **JP0399**
Premier Locator+: **JP0399SS**

Java 1.2 Books are Here!
Save 20% on the Best Java Books
Buy any book and save 20% when you order through Marketplace on The Development Exchange. Not only do we have the best Java books, we also carry the latest titles on Web design and development, ActiveX, enterprise development and more! Plus you can get 40% off our Featured Books of the Week. Select from our top titles for great savings—but act quickly because each week means another set of titles at incredible discounts!
Order online at www.devx.com
DEVX MarketPlace
FTP Books, 209 Hamilton Ave., Palo Alto, CA 94301 • Tel: 650-833-7100 • Fax: 650-853-0230
Java is a trademark of Sun Microsystems. ActiveX is a registered trademark of Microsoft Corporation.
The Development Exchange is a trademark of Fawcette Technical Publications.

Solaris Ups the Ante

Fast thread synchronization and a state-of-the-art memory system bring top-tier Java performance to your Sun workstation

by Paul Tyma



Paul Tyma is president and chief scientist at preEmptive Solutions Inc., a Java training and consulting firm. Paul is the lead author of the book *Java Primer Plus* and a frequent conference speaker on Java Performance. He is currently a Ph.D. candidate in computer engineering at Syracuse University and lead architect for preEmptive's DashO Java bytecode optimizer/obfuscator. Reach him at ptyma@preemptive.com.

It has often occurred to me that the company that created Java often seems to lag behind in pioneering runtimes. Granted, Hotspot has been in the news for ages, but it has yet to be released. Well, it turns out that Sun's Solaris Software Group wasn't just watching the world go by. The next generation of the JDK 1.2 for Solaris (Production Release) is going to look a bit different under the hood. It now includes a new JIT compiler with some pretty interesting (if not bleeding edge) features. After doing some benchmarking, I'm impressed. This new VM is looking pretty spiffy.

Sun's official word on this is, "The JDK for Solaris is geared toward delivering the best possible performance and scalability to server-side Java applications found in the Solaris enterprise environment. As such, it includes an enhanced Java Virtual Machine (JVM) with an optimizing Just-In-Time (JIT) compiler, fast thread synchronization, and state-of-the-art memory system (with exact, generational garbage collection) all of which serve to significantly increase scalability and improve performance."

The enhanced memory system isn't just a state-of-the-art collector, it allocates memory in a flash too. The final Production Release is slated for early 1999, and you can grab the beta now from the Sun at www.sun.com/solaris/jdk/download.1.2.

I was pleasantly surprised by the help I got from Sun to test the VM. Although I'm sure they didn't divulge all of their secrets, they gave me access to prerelease versions and the engineers themselves. Besides my normal onslaught of empiri-

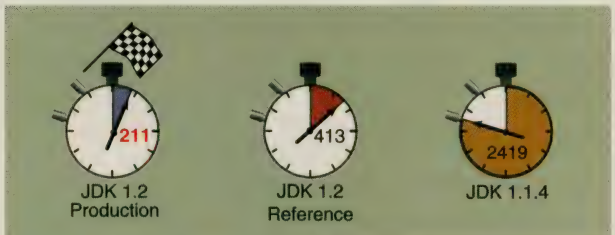


Table 1 Matrix Multiplication The new Production Release runs twice as fast as the 1.2 Reference Implementation, and both of these leave JDK 1.1.4 far behind. Time is shown in milliseconds.

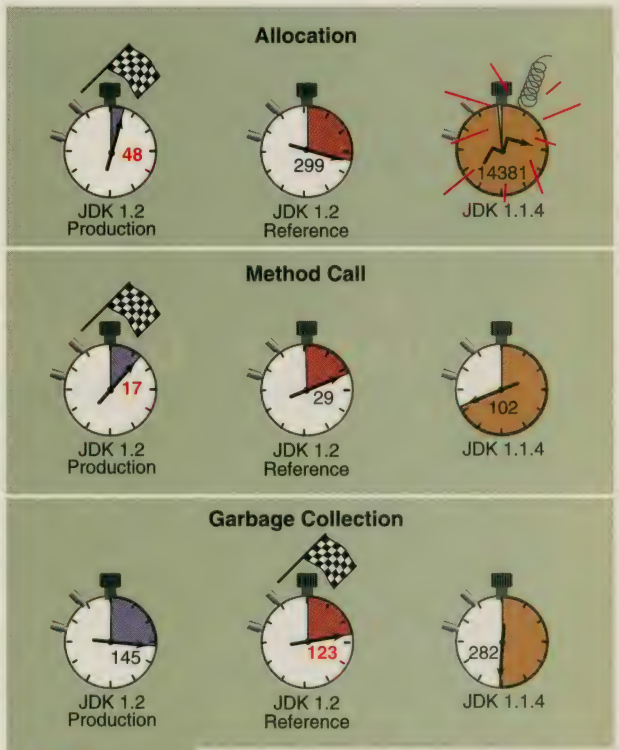


Table 2 Memory The new VM is 300 times faster at the allocation test than JDK 1.1.4.! Time is shown in milliseconds.

FastJ™

Java™ code in embedded? Yes!

Introducing the Industry's First Highly Optimizing Cross-Compiler Suite for the Java Language

With the FastJ™ compiler suite, you get all the benefits of the Java programming language without the overhead of a Java Virtual Machine. Write your program in Java source code. Compile it. Straight to machine code. For the most popular 32-bit embedded processors. PowerPC, 68K/CPU32, ColdFire, and M•CORE.

What's more, FastJ is built on top of Diab Data's industry leading optimization and code generation technology. With FastJ, you get the fast, compact, stable code you want with the ability to mix C/C++ and assembly. And with Diab Data's award-winning RTA Suite, you get the tools to analyze and improve program performance, detect run-time errors, and get moving to the next winning design.

DIAB DATA
Defining Compiler Performance
An Integrated Systems Company

USA: Tel: 650.571.1700, Fax: 650.571.9068. Europe: Tel: +49 (0)89.9393.1191, Fax: +49 (0)89.930.5184

© 1998, Diab Data, Inc. All Rights Reserved. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States and other countries. Diab Data, Inc. is independent of Sun Microsystems, Inc. FastJ is a trademark of Diab Data, Inc.

Check out our white paper to learn more about compiling Java for embedded systems.

www.ddi.com info@ddi.com

Solaris JDK 1.2 beta Production Release Benchmarks

Because most of the VMs tested in previous columns were benchmarked on Windows boxes, we can't objectively compare them to these scores. The test machine for this benchmark was a dual CPU 200-MHZ Ultra 2, with 256 MB of RAM, running Solaris 2.6.

First we ran the regular stew of benchmarks with which I've tortured all other VMs. In order to get some feeling for what this new VM is buying us, I also grabbed the JDK 1.1.4 and JDK 1.2 (all on Solaris, of course) for comparison.

The JDK 1.1.4 is presented here to show "where we came from." The JDK 1.2 is the reference system that came from Sun's Java Software Group (what we knew as JavaSoft). It contains a JIT but it is otherwise a completely different animal. The Solaris Software Group's VM is why we're here, and for these tests we'll tag it as the JDK 1.2 Production Release. Apparently the difference between the two JITs (and groups within Sun) is that Java Software has produced the Reference Implementation, which isn't tuned for a specific platform. Solaris Software produces the Production Release, which is tuned for Solaris. For more information, see http://java.sun.com/products/jdk/faq.html.

In the Matrix and Memory tables, we see some dramatic results. Even the JDK 1.2 Reference spanks the JDK 1.1 without mercy. The new VM takes things a step further and often runs twice as fast as the 1.2 Reference. Comparing to the 1.1 JDK to the new VM is almost silly, it's 300 times faster at the allocation test than JDK 1.1.

As I always warn, the GC timing is nothing short of dangerous because it's induced with the `System.gc()` call. This is not a foolproof method of getting garbage collection to run, and as more complex GC systems evolve, its behavior is assured to become even less predictable. This test also creates objects in a uniform (yet continuous) manner. All the objects live the same amount of time providing an inappropriate test for a generational system. Nevertheless, Sun's new VM fairs well, but it should be significantly more useful in a real environment.

The results in the API table don't exceed expectation. Some goals of the new VM were fast memory allocation and fast synchronization. The benchmarks shows they weren't kidding around on memory allocation. The synchronization piece is probably responsible for the API advantage also. All gets/puts are synchronized calls.

I whipped up a trivial example to test synchronized method-call overhead, and in the Method Call table we see the abilities of this new VM once again. As you can see, although synchronization isn't free, it is far less expensive in the new VM.

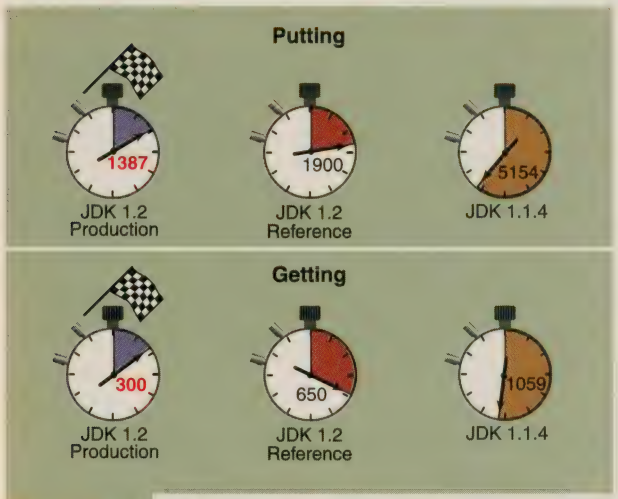


Table 3 API All gets/puts are synchronized calls, so the new VM's fast synchronization really pays off. Time is shown in milliseconds.

cal tests, the information presented here is straight from the horse's mouth (or straight from the information provided by the development version that "showed" me what it was doing).

Being Exact

One of the most tantalizing features of this VM is its non-conservative garbage collector (GC), that is—it is "exact" or "precise." Most GC algorithms are conservative. They can't always be sure whether a possible object reference is truly an object or maybe just an integer that happens to have a value pointing to valid object space. Conservative collectors don't have rock-solid information as to what objects exist (freaky but true). Therefore, when such a dilemma presents itself, a conservative collector has to be safe and assume it's a reference. An exact collector isn't so naïve: It knows exactly where all the objects in the system reside, and it never has to worry about a possible mix-up. (It doesn't need to play conservative, it knows it's right.)

The GC is also generational. Rightfully or not, Generational GC has become somewhat synonymous with state-of-the-art in the Java community. The GC in the Solaris JDK has two generations, each with their own GC algorithm. The younger generation uses semi-space (stop-and-copy) and the older generation uses mark-compact. In generational GCs, the youngest generations get collected more frequently, older objects are moved to the older generation and collected less frequently. This VM definitely sports the most complex, interesting, and best performing GC we've seen to date.

It also has mixed-mode operation. That is, it doesn't always JIT compile code. It is fully capable of running heavily used code after it's been JIT compiled and branching into seldom-used code in interpreted mode. To top it off, it has the ability to optimize code for further performance enhancement.

Optimizations

Merely JIT compiling Java bytecode buys you a solid speed up over any interpreter, but that's just the beginning of the benefits. Once the code is JIT compiled, you have the ability to perform a variety of optimizations on it. Keep in mind, optimization is not easy: Amazing nuances come into play making even cookbook code transformations delicate instruments. However, you have a great deal more optimization flexibility.

The JIT compiler implements several optimizations that offer a high return on your investment (either because they are easy to implement, have profound effects or both). The optimizations include method inlining, extended basic block-value numbering, common-subexpression elimination, array bounds-check elimination, and virtual-method inline caching. These terms can be rather broad, and the extent to which each works might not have been clear, except that the development version of the VM I was given let me "see" what it was going to do. (Unfortunately, shipping versions of the VM won't provide this ability for the general public).

Java GUIs tend to be just that—gooey

Method inlining was quite aggressive. It's a classic optimization, and I won't waste time heralding its virtues here, but a smart implementation is always a winner. Value numbering and common-subexpression elimination are ways of removing redundant calculations within code, as in this example:

```
Y = (X * 10) + Z;
Z = A + (X * 10);
```

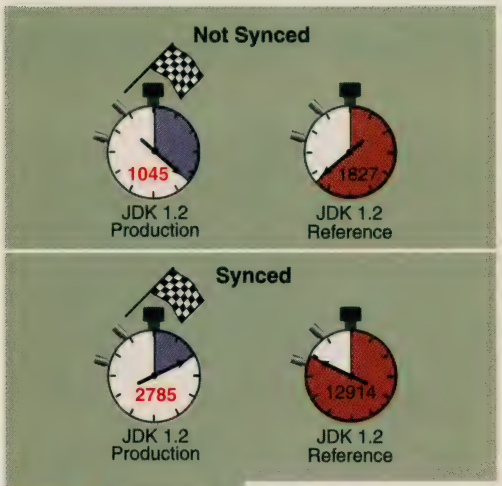


Table 4 Method Calls Although synchronization isn't free, it is far less expensive in the new VM. Time is shown in milliseconds.

Obviously, once we calculate `(X*10)`, there is no need to do it again. Typically, redundant calculations are much less apparent to the naked eye.

Virtual-method inline caching is a great idea that I'm glad to see alive and well. For example, you could have an abstract `Shape` class (with an abstract `draw()` method) that is subclassed by Rectangle, Triangle, and Circle. Then if another method has this code:

```
Shape S = getSomeActualShape();
S.draw();
```

Any sort of inlining is potentially dangerous because we don't know what the shape is. However, if we decide (through profile information, a greedy algorithm, or black magic) that it is usually a Rectangle, we'll inline the Rectangle's `draw()` method. If during any particular execution we find it's not a Rectangle, we call the original code.

Instead of the `S.draw()` code, you can use this code:

```
if (S instanceof Rectangle) {

// inlined Rectangle Draw code
// draw 4 lines

} else S.draw();
```

Fact Sheet

Performance: Fast (with memory blazing)

Purity: 100% (it IS the JDK!)

Runs on: Solaris (Sparc and x86)

Execution Method: Mixed-mode JIT/interpreter

Garbage Collector: Exact generational, with semi-space and mark-compact for the generations

Availability: Early 1999 as the new Solaris JDK. Beta available at www.sun.com/solaris/jdk/download.1.2/

It's not always quite that kludgy, but if `S` really *were* a Rectangle most of the time, the above code would be a win. The VM is also smart enough to realize that maybe only Rectangle was ever loaded, simplifying the problem altogether.

Solid GUIs

We can't go comparing Symantec's or Microsoft's VM in this environment. Stepping across architectures and operating systems makes too far a leap to get meaningful data. However, I can say real-world apps running in the new VM were quite spry—the subjective "feel" was as good as or better than Microsoft's (my previous favorite). Java GUIs tend to be just that—gooey, sorta squishy feeling, but not with this Solaris VM—everything was crisp and responsive.

The improvement of the new runtime versus its JDK 1.1 counterpart is nothing short of astonishing in many cases. I don't know if that means the new one is that good or the original JDK was that bad. The magic Sun performed for memory allocation and garbage collection worked, and the scores (and relative improvements) are awe inspiring.

Special thanks to Ross Knippel and Ole Agesen of Sun for getting me access to the VM and answering vats of questions, and thanks to Doug Lea for the use of his machine.

TechTIP

Arrays are Cheaper and Safer than Vectors

In Java, you can use either a Vector or a one-dimensional array to store an ordered collection of objects. The difference between the two types of storage is that the array allocates storage for a fixed number of elements where as the Vector can grow dynamically. Another difference is that an array can be used to store primitives (int, boolean, and so on) where as a Vector can only store pure objects. You should also keep in mind that arrays are type-safe. An array can only store objects of a particular class, and only objects of that class can be retrieved from it. Assigning that object to a variable of its type requires no explicit casting. For example:

```
SomeObject[] soArray = new SomeObject[10];

// Fill array with some object
for (int i = 0; i < 10; i++)
  soArray[i] = new SomeObject(i);

// Get objects from the array
for (int i = 0; i < 10; i++)
  SomeObject so = soArray[i];
```

The variable `soArray` only holds variables of the type `SomeObject`. The storage for ten references for objects of the type `SomeObject` is allocated during compilation.

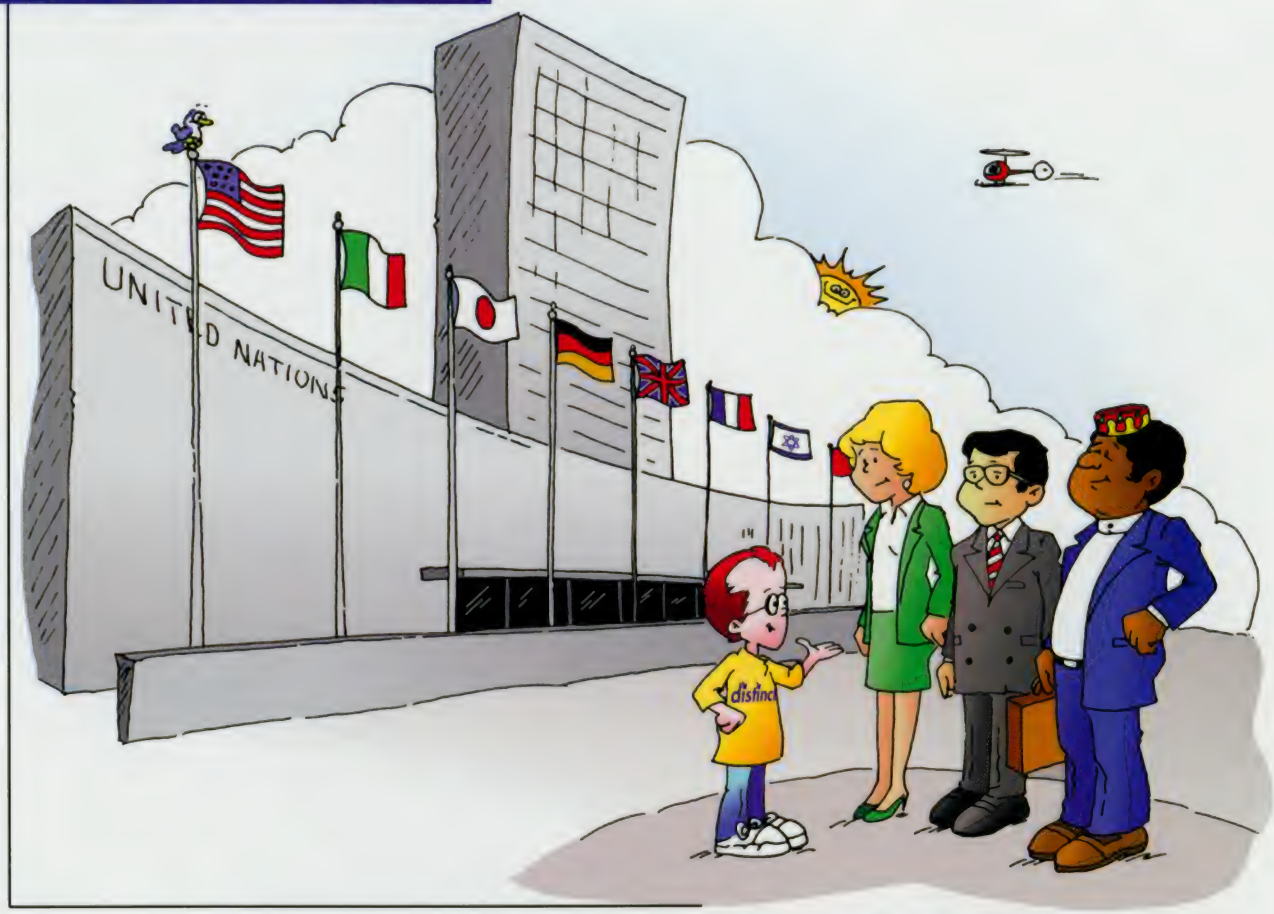
A Vector stores objects of the class `Object`. You can use a Vector to achieve the same functionality that you can achieve with an array:

```
Vector soVector = new Vector();

// Fill Vector with some object
for (int i = 0; i < 10; i++)
   soVector.addElement(new SomeObject(i));

// Get objects from the array
for (int i = 0; i < 10; i++)
  SomeObject so = (SomeObject)(soVector.elementAt(i));
```

The `elementAt()` method returns a reference to an instance of an `Object`. This `Object` has to be explicitly cast to `SomeObject`. If the object stored in `soVector` is not of the type `SomeObject`, the program will throw a ClassCastException at run time. Since explicit casts are expensive, arrays should be the preferred means of storage if the type of the object to be stored is known at compile time and if you can predetermine the amount of storage.

—Ajit Sagar

Doing it right with The Cody Kid.™



"Your heterogenous environments are a lot like ours."

Develop, debug and deploy cross-platform network applications fast, with Distinct RPC Toolkits.

Introducing the Distinct ONC RPC/XDR Toolkits, the fastest way to bring your corporate applications to the Windows 95 and NT desktop. In addition to our toolkit for C and C++ developers, Distinct delivers the first Java-based RPC toolkit. Distinct ONC RPC/XDR Toolkit for Java puts RPC-based distributed programming right at Java programmers' fingertips. And because ours is the first toolkit of its kind to receive 100% pure Java certification, you can now write platform-independent distributed applications. So, whether you are porting the front end of your UNIX applications to the Windows platforms, or creating a completely new distributed application over a heterogeneous network, rely on Distinct RPC toolkits to help you do it faster and more reliably. It's also the easiest way ever devised to bring your UNIX-based applications to your desktop.

Our RPC Toolkits are backed by comprehensive technical support from Distinct, a pioneer in enabling technologies for software professionals since 1984. To get your free trial copy, visit the Cody Kid™ on our Web site today.

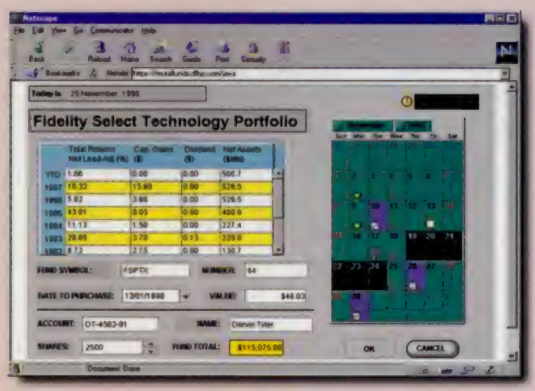
distinct®
Develop. Debug. Deploy.™

Download a trial version now at www.distinct.com or call 408-366-8933

DISTINCT is a registered trademark and the DISTINCT logo, product names, the DEVELOP. DEBUG. DEPLOY. tagline, DOING IT RIGHT WITH THE CODY KID., and THE CODY KID are trademarks of Distinct Corporation. Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc., in the United States and other countries. Windows and Windows NT are either trademarks or registered trademarks of Microsoft Corporation. All other trademarks are the property of their respective owners. ©1998 Distinct Corporation. 003

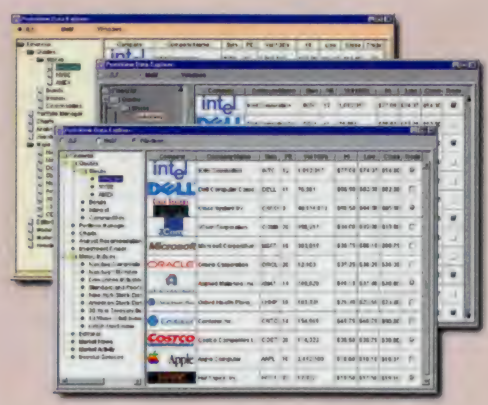
Build It - Once

Build It - Right



ProtoView JFCSuite™

Powerful, extendable, lightweight components built on the foundation of JFC, the ProtoView JFCSuite contains JFCDataCalendar, JFCDataExplorer and JFCDataInput. JFCDataCalendar has drop-down date selection, multiple date & range selection and day-specific font/color/image properties. JFCDataExplorer provides an intuitive explorer UI by combining the JFC JTree and JTable into one integrated and enhanced package. Right-hand views can be customized to display JavaBeans components for each branch of the tree. JFCDataInput provides advanced validation and formatting for Currency, Date, Mask Edit, Numeric and Password input. Together, these components have everything you need to build solid, state-of-the-art, JFC based Java applications.



ProtoView JSuite (AWT 1.1 JavaBeans)™

Winner of the JDJ World Class Award. The JSuite includes the DataTableJ grid component with JDBC and Visual Café database support. Also includes TreeViewJ, CalendarJ, TabJ and WinJ (MaskEdit, Currency, Numeric Edit, DateEdit Long, Time, Combo Box and many more).

ProtoView Java Components —— Download Free Trial Versions

Order Online Today!

www.protoview.com
1-800-231-8588

PROTOVIEW™
Software Tools for Component Based Development

Copyright © 1998 ProtoView Development Corporation All rights reserved All ProtoView product names are trademarks of ProtoView Development Corporation

YES ☑ Send me one year (13 issues) of *Java™ Pro* for only $29.95

Name _____

Address _____

City _____ State _____

Zip _____ Country _____

E-mail address _____

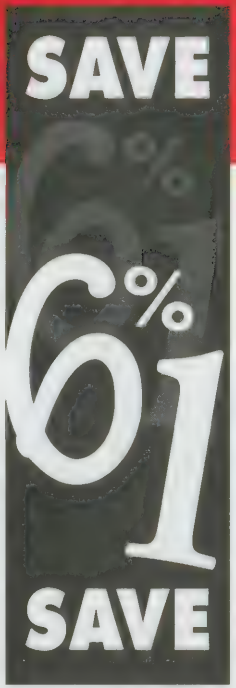
❑ Bill me ❑ Payment enclosed Charge my: ❑ Visa ❑ Master Card ❑ AMEX

Card # _____

Expiration date _____

Signature _____

One year is 12 issues plus one special issue. Savings based on annual cover price of $77.35. Please allow 4-6 weeks for delivery of your first issue. International subscriptions must be payable in US dollars plus postage: Canada/Mexico add $18/year for surface delivery; all other countries add $44/year for airmail delivery. Canadian GST included. 4040



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 317 Palo Alto, CA

POSTAGE WILL BE PAID BY ADDRESSEE

P.O. Box 54584
Boulder, CO 80323-4584

JAVA Pro Savings Voucher

Get Java™ Pro for only $2.30 per issue—and save over 61% off the cover price. A single tip in one issue can more than pay for a year's subscription. Order Now and Save!....

Annual Cover Price	Your Price	Subscription Savings
$77.35	$29.95	61%

❑ **1 Year** (13 issues) $29.95—SAVE 61%

NAME _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

❑ Payment enclosed ❑ Bill me later

My Guarantee:
If at any time I do not agree that *Java™ Pro* is the best, most useful source of information on Java Programming, I can cancel my subscription and recieve a full refund.

One year is 12 monthly issues and one special issue. Savings based on annual cover price of $77.35. Please allow 4-6 weeks for delivery of first issue. International subscriptions must be payable in U.S. dollars plus postage. Canada and Mexico add $18/year, all other countries add $44/year for airmail delivery. Canadian GST included.

4039

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 317 Palo Alto, CA

POSTAGE WILL BE PAID BY ADDRESSEE

P.O. Box 54584
Boulder, CO 80323-4584

Become a Charter Subscriber NOW!

(And You Won't Miss a Single Issue)

JAVAPro™

YES ☑ Send me one year (13 issues) of JAVA Pro for just $29.95

Name _____

Address _____

City _____ State _____ Zip _____ Country _____

E-mail address _____

❑ Bill me ❑ Payment enclosed Charge my: ❑ Visa ❑ Master Card ❑ AMEX

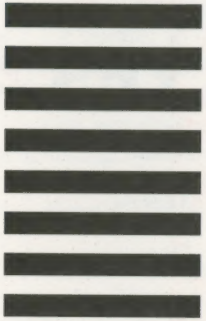
Card # and expiration date _____

Signature _____

One year is 12 issues plus 1 special issue. Annual cover price is $77.35. Make checks payable to JavaPro. Please allow 4-6 weeks for delivery of first issue. International subscription must be payable in U.S.Dollars plus postage. Canada/Mexico add $18/year; all other countries add $44/year for air mail. Canadian GST included.

4038

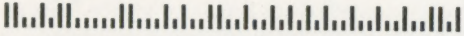
NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 317 Palo Alto, CA

POSTAGE WILL BE PAID BY ADDRESSEE

JAVA™Pro

PO BOX 54584
BOULDER CO 80323-4584